2014 IEEE International Conference on Data Mining Workshop

Emotion Recognition from Text Based on Automatically Generated Rules

Shadi Shaheen
Computer Science Dept.
American University of Beirut, Lebanon
sis15@mail.aub.edu

Wassim El-Hajj
Computer Science Dept.
American University of Beirut, Lebanon
we07@aub.edu.lb

Hazem Hajj
Electrical and Computer Eng.
American University of Beirut, Lebanon
hh63@aub.edu.lb

Shady Elbassuoni
Computer Science Dept.
American University of Beirut, Lebanon
se58@aub.edu.lb

***Abstract*—With the growth of the Internet community, textual data has proven to be the main tool of communication in human-machine and human-human interaction. This communication is constantly evolving towards the goal of making it as human and real as possible. One way of humanizing such interaction is to provide a framework that can recognize the emotions present in the communication or the emotions of the involved users in order to enrich user experience. For example, by providing insights to users for personal preferences and automated recommendations based on their emotional state. In this work, we propose a framework for emotion classification in English sentences where emotions are treated as generalized concepts extracted from the sentences. We start by generating an intermediate emotional data representation of a given input sentence based on its syntactic and semantic structure. We then generalize this representation using various ontologies such as WordNet and ConceptNet, which results in an emotion seed that we call an emotion recognition rule (ERR). Finally, we use a suite of classifiers to compare the generated ERR with a set of reference ERRs extracted from a training set in a similar fashion. The used classifiers are k-nearest neighbors (KNN) with handcrafted similarity measure, Point Mutual Information (PMI), and PMI with Information Retrieval (PMI-IR). When applied on different datasets, the proposed approach significantly outperformed the existing state-of-the-art machine learning and rule-based classifiers with an average F-Score of 84%.**

Keywords—Emotion Recognition from Text; Natural Language Processing; Data Mining;

I. INTRODUCTION

Recognizing user's emotions is a major challenge for both humans and machines. On one hand, people may not be able to recognize or state their own emotions at certain times. On the other hand, machines need to have accurate ground truth for emotion modeling, and also require advanced machine learning algorithms for developing the emotion models. Hard sensing methods and soft sensing methods have been traditionally used to recognize user's emotions. With hard sensing methods, sensors provide the data sources that may be relevant to emotion recognition such as audio, gestures, eye gazes and brain signals [1-4]. Additional sensors may be attached to the user to provide personal physiological cues such as heart rate sensors. However these wearable sensors are not applicable in practical and natural settings since they can be obtrusive to the user. Soft sensing methods, on the other hand, extract information from software that already exists with the user (on her phone or PC) and analyzes it for the purpose of recognizing the user's emotions. Examples of software that can be analyzed to classify user's emotions include calendar, email, desktop activity, and social networking interactions. In this work, we focus on classifying emotions from text as text is not obstructive and is considered the main tool of communication between people and machines [5]. A market analysis done by OFCOM in the UK shows that 68% of the people tend to communicate with family and friends using text-based communication, while 49% use face-to-face communication.

Emotion recognition from text has many applications. Consider for example an employee sending a harsh email to his colleague or superior. A tool that can analyze the email for emotions and alert the employee about its harshness before sending it comes in very handy to protect the employee's state. Consider also an emotion-based search engine that ranks documents according to the emotion requested by the user. Such an engine could prove to be very beneficial to users in a certain emotional state and can improve the effectiveness of the information retrieval process. Other useful tools that can benefit from emotion recognition from text include recommender systems that aim to personalize recommendations based on the user's emotions.

Several emotional models have been used when building emotion recognition systems [6, 7]. A recently suggested model is the hourglass model [6], which is biologically-inspired, psychologically-motivated, and based on the idea that emotional states result from the selective activation/deactivation of different resources in the brain. Another common model is Ekman's model [7] which categorizes the emotions into six universal categories.

In this paper, we aim to recognize the six emotions suggested by Ekman: happiness, sadness, anger, fear, disgust and surprise [7]. We reduce the problem of emotion recognition or emotion detection from text to the problem of finding relations between the input sentence and the emotional content within it. Intuitively, finding these relations relies on discovering specific terms (emotional keywords, verbs, nouns, etc.) in the input sentence and other deeper inferences that are related to the meaning of the sentence. Once these terms and their relation to the meaning of the sentence are found, they can be generalized and considered as emotion recognition rules (ERRs). For example, consider the sentence "I received many gifts on Christmas Eve"; Assuming that this sentence reflects a happy emotion, by analyzing the sentence (more details in later sections) we can reach to the conclusion that the verb

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.80

"received" and the noun "gifts" are the most important parts of the sentence, and consequently we can come up with a rule that says "*receiving gifts*" reflects the emotion *happy*.

According to this analysis and unlike previous work, we developed our system to be context sensitive by performing deep semantic and syntactic analysis using various NLP tools. Our system stands out among previous work, as it incorporates the idea of concepts in the research of emotion mining from text. It also uses existing training data and the World Wide Web to enhance its classification accuracy by utilizing various measures such as pointwise mutual information (PMI) and pointwise mutual information with information retrieval (PMI-IR). In addition, our model is flexible enough and can be used to detect any number of emotion classes for which some training data exists.

We tested the validity of our model using a set of experiments on real-world datasets. In one experiment, the proposed method was trained on a dataset extracted from Twitter and tested on another completely different dataset based on blogs. The achieved F-score was 84%, which according to our knowledge outperforms the current state-of-the-art methods in emotion recognition from text (an increase of around 10% in F-score measure).

Our main contributions can be summarized as follows:

- we perform deep syntactic and semantic analysis of sentences using various NLP tools combined with a set of linguistic rules to generate a concise emotion lexicon,
- we utilize ontologies such as Wordnet and ConceptNet to generalize our lexicon,
- we develop a new similarity metric that can be used to classify a given sentence into one of various emotional classes, and
- we use a suite of classifiers to detect emotions of sentences with very high precision and recall.

The rest of the paper is organized as follows. Section 2 presents the literature review on emotion recognition from text. Section 3 describes the datasets we used in building and testing our system. Section 4 presents our proposed methodology for emotion mining from text. Section 5 presents our experiments and the achieved results and section 6 concludes the paper.

II. RELATED WORK

Emotion classification can be divided into two different categories: coarse-grained and fine-grained classification. Classifying emotions on a coarse-grained level (positive or negative) can be accurately perceived from text. Hancock et al. [8] used content analysis Linguistic Inquiry and Word Count (LIWC) to classify emotions as positive or negative. They found that positive emotions are expressed in text by using more exclamation marks and words, while negative emotions are expressed using more affective words. However, this method is limited to positive/negative emotions (happy vs. sad).

On the other hand, classifying emotions on a fine grained level (for example, the six Ekman emotions) requires semantic and syntactic analysis of the sentence and can be done using three methods: (1) Keyword-based detection, (2) Learning-based detection, and (3) Hybrid detection. We discuss each family of methods separately.

A. Keyword-based detection

Here, classifying emotions is done by searching for the emotional keywords in the input sentence [7]. Early work on understanding emotion expression in text was done by Osgood et al. [9, 10]. They used multidimensional scaling for visualizing the affective words in order to compute similarity ratings between them. The dimensions used by Osgood were "evaluation", "potency" and "activity", where evaluation quantifies how much a word refers to a pleasant or unpleasant event, potency quantifies the emotional intensity of a word (strong or weak), and activity refers to whether a word is passive or active.

Strapparava et al. developed a linguistic resource for lexical representation of affective knowledge named WordNet – Affect [11]. WordNet-Affect contains a subset of synsets that represent affective concepts corresponding to affective words. Emotion Classification is then done by mapping emotional keywords that exist in the input sentence to their corresponding WordNet-Affect concepts.

However, classification methods based on only keywords suffer from (1) the ambiguity in the keyword definitions in the sense that a word can have different meanings according to usage and context, (2) the incapability of recognizing emotions within sentences that do not contain emotional keywords, and (3) the lack of linguistic information.

B. Learning-based detection

In machine learning methods, the emotion is detected by using classification approaches based on a training dataset. Strapparava et al. [12] developed a system that used several variations of Latent Semantic Analysis to identify emotions in text when no affective words exist. However their approach achieved a low accuracy because it is not context sensitive and lacks the semantic analysis of the sentence.

Burget R. et al. [13] proposed a framework that depends heavily on the pre-processing of the input data (Czech Newspaper Headlines) and labeling it using a classifier. The pre-processing was done at the word and sentence levels, by applying POS tagging, lemmatization and removing stop words. Term Frequency – Inverse Document Frequency (TF-IDF) was used to calculate the relevance between each term and each emotion class. They achieved an average accuracy of 80% for 1000 Czech news headlines using SVM with 10-fold cross validation. However their method was not tested on English dataset. Also it is not context sensitive as it only considers emotional keywords as features.

Dung et al. [14] exploited the idea that emotions are related to human mental states which are caused by some emotional events. This means that the human mind starts with initial mental state and moves to another state upon the occurrence of a certain event. They implemented this idea using Hidden Markov Model (HMM) where each sentence consists of multiple sub-ideas and each idea is considered an event that causes a transition to a certain state. By following

the sequence of events in the sentence, the system determines the most probable emotion of the text. The system achieved an F-score of 35% when tested on the ISEAR dataset [15] (International Survey on Emotion Antecedents and Reactions), where the best precision achieved was 47%. The low accuracy was mainly due to the fact that the system ignored the semantic and syntactic analysis of the sentence, which made it non-context sensitive.

C. Hybrid detection

In hybrid methods, emotions are detected by using a combination of emotional keywords and learning patterns collected from training datasets, in addition to information from different sciences, like human psychology [16]. Few works addressed the problem of extracting emotions from text that does not contain emotional keywords [16-19].

Wu et al. [16] proposed a novel approach for sentence level emotion mining based on detecting (1) predefined semantic labels and (2) attributes of the sentence, then classifying emotions based on psychological patterns of human emotions called emotion generation rule (EGR). However, their approach was limited to one emotion (happy) since the method exhibited a lot of ambiguity when one EGR can generate more than one emotion.

Cheng-Yu Lu et al. [17] presented vent-level textual emotion sensing by building a mutual action histogram between two entities where each column in the histogram represented how common an action (verb) existed between the two entities. They achieved an F-score of 75% when tested on four emotions. However, their method does not consider the meaning of the sentence and is highly dependent on the structure of the training data, i.e. the grammatical type of sentences in the training data and the frequency of the emotions for a certain subject. Moreover, only four of the six Ekman emotions are used in the classification.

F. Chaumartin [18] developed a linguistic rule-based system UPAR7, using WordNet [20], SentiWordNet [21] and WordNet-Affect [11] lexical resources. The system makes use of the dependency graph obtained from the Stanford POS tagger [22], where the root of the graph is considered the main subject. Each word in the sentence is rated individually for each emotion. Then the rating of the main subject (main word) is boosted, as it is more important than the rest of the words in the sentence. The best-achieved accuracy of this approach was 30% for the six emotions of the Ekman model. In addition to the low accuracy of this approach, it is not context sensitive and lacks the global understanding of the sentence.

Yang et al. [23] proposed a hybrid model for emotion classification that includes lexicon-keyword spotting, CRF-based (conditional random field) emotion cue identification, and machine-learning-based emotion classification using SVM, Naïve Bayesian and Max Entropy. The results generated from the aforementioned techniques are integrated using a vote-based system. They tested the system on a dataset of suicide notes where it achieved an F-score of 61% with precision 58% and recall 64%. This method achieved relatively good results; however, both the classifier and the dataset are not available.

Ghazi et al. [24] tried hierarchical classification to classify the six Ekman emotions. They used multiple levels of hierarchy while classifying emotions by first classifying whether a sentence holds an emotion or not, then classifying the emotion as either positive or negative and finally classifying the emotion on a fine-grained level. For each stage of classification they used different features for the classifier, and they achieved a better accuracy (+7%) over the flat classification where flat classification is classifying the emotions on a fine-grained level directly. The main drawback of this approach is that it is not context sensitive.

Neviarouskaya et al. [19] developed EmoHeart, a lexical rule-based system that recognizes emotions from text and visualizes the emotion expressions in a virtual environment. Their system is used in the game Second Life [25]. The system starts by looking for emotional abbreviations and emoticons. If not found, it processes the sentence on different levels (word level, phrase level and sentence level) to generate an emotional vector of the sentence, where each element in the vector represents an emotional class intensity. At word level, each word in the sentence is mapped to its emotional vector, where they manually build a dataset of emotional vectors for many words. At the phrase and sentence levels, they combine the emotional vectors collected from the words by either performing summation or maximization among the vectors. The emotion of the sentence is the maximum intensity of the vector. They achieved an average accuracy of 75% when tested on a manually annotated dataset. However this method exhibits few drawbacks. First, the system does not handle the case when negation exists in the sentence. Second, it is based on an affective database where emotion categories and intensities were assigned manually to each word in the database, which makes their approach hard to extend to classify more emotions. We consider this method the state-of-the-art given the high accuracy it achieves and we compare our approach to it achieving a significant increase of around 10% in F-score.

III. DATASET

Our approach makes use of two annotated datasets where each sentence is annotated with one of the six Ekman emotions. Aman 2007 [26] is the first dataset we used which is composed of emotion-rich sentences collected from blogs and annotated with the six Ekman's emotion labels. We decided to choose this dataset as blog posts offer variety in writing styles, choice and arrangement of words and topics.

TABLE I. AMAN DATASET SPECIFICATION

Emotion	Symbol	#Posts	#Sentences
Happiness	Hp	34	848
Sadness	Sd	30	884
Surprise	Sp	31	847
Disgust	Dg	21	882
Anger	Ag	26	883
Fear	Fr	31	861

It is a public dataset annotated by two judges (details in Table 1). In one of our experiments, we used part of this dataset for training and the other part for testing.

Given the popularity of social media and their richness in opinion and emotion contents [27], we used sentences collected from twitter as our second dataset. Each tweet is annotated with one of five emotions: happiness, sadness, surprise, anger and fear (5 of the 6 Ekman emotions are included in this dataset). The annotation is done according to the hashtags of each tweet [28]. For example, the emotion of the sentence is considered to be "anger" if one of the anger hash tags was found in the sentence (i.e. #irritation, #annoyance, #irritating). The full dataset contains 2,488,982 tweets, but we only extracted and used 18,000 tweets, 3,600 belonging to every emotion category. This dataset was used in another experiment where it was solely used for training. For testing we used the Aman dataset. The achieved results were very promising (almost 10% better than the state-of-the-art). This experiment highlights the strengths and robustness of our approach where we trained using one dataset and tested using a completely different dataset.

IV. METHODOLOGY

In our approach, we propose a novel approach for emotion classification in English sentences where the emotions are treated as concepts extracted from the sentence. Concepts can be expressed as nouns, adjectives, adverbs, and verbal phrases or as a combination of different phrases. For example, consider the sentence "I found a solution to a problem". This sentence represents an emotional concept extracted from the semantic relations between its words. The sentence indicates the emotion "Happiness", as the concept of solving a problem will trigger the emotion "Happiness". Our representation of concepts depends on two principles.

The first one is the compositionality principle [19], which states that in a meaningful sentence, if the lexical parts (meaningful words) are taken out of the sentence, what remains are the rules of composition. If we applied this principle on the sentence "I found a solution to a problem", once the meaningful lexical items are taken away ("found", "solution" and "problem"), what is left is the pseudo-sentence "I F S to P" (F for found, S for solution, and P for problem). The task of understanding the sentence becomes a matter of describing the relation between "F", "S" and "P".

The second principle states that in order to understand a sentence, syntactic and semantic analysis should be performed [29]. Performing syntactic analysis on the sentence reflects the structure of the sentence. It involves determining the part of speech of each word, for example noun, verb, adjective, adverb, etc. Performing semantic analysis on the sentence reflects the relations between its words. For example, performing syntactic analysis on the sentence "I found a solution to a problem" will result in "PRP VB NN TO NN", where PRP means pronoun, VB means verb, TO means to, and NN means noun. Performing semantic analysis, on the other hand, will reflect the relation between the verb "found" and the words "I, solution, problem"; "I" is the subject of the verb "found" while "solution" is the object and "problem" is related to the verb "found" with the preposition "to".

The syntactic and semantic analysis of a given sentence is done by first constructing the dependency tree of the sentence and then applying rules that aim to prune the tree keeping only the subtree that represents the emotion in the sentence. The intuition behind this pruning step is to build a lexicon of emotion-related phrases that are general enough and can later be used to detect the emotion of any given input sentence. The output of the syntactic and semantic analysis just mentioned is translated in our model to an intermediate representation of the sentence that we call Emotion Recognition Rule (ERR). ERRs are composed of four types of constructs: (1) verb-noun clauses (VNCs), (2) noun clauses (NNs), (3) adjectives (JJs), and adverbs (RBs). We chose only these four types of constructs as they reflect actions and event descriptions. Typically, emotions can be caused either by a certain action or when describing a certain incident. Figure 1 shows a sample dependency tree extracted from the following sentence: "It was the best summer I have ever experienced". "experienced" is the main verb; "I" is related to "experienced"; "best" is related to "summer" and "best summer" is related to "experienced".

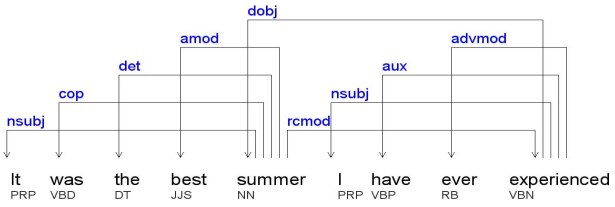

Fig. 1. Dependency Tree

Figure 2 shows a sample ERR extracted from the same sentence, however it only holds the emotional part of the sentence and the relation between its words. Our pruning strategy will be discussed in more detail later in this section.

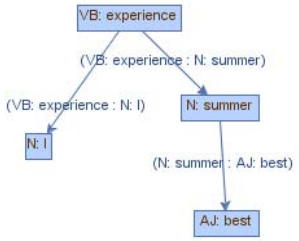

Fig. 2. ERR representaion extratced from the dependency tree in Fig. 1

Our methodology is composed of two main phases: (1) an offline phase - building a reference set of ERRs, and (2) a comparison and classification phase.

A. *Offline phase*

The goal of this phase is to build a set of annotated reference ERRs. To build our reference set, we need a dataset manually annotated with one of the six Ekman emotions. We process each annotated sentence in the dataset by the Sentence Processor module to generate its corresponding ERR. Figure 3 summarizes this phase.

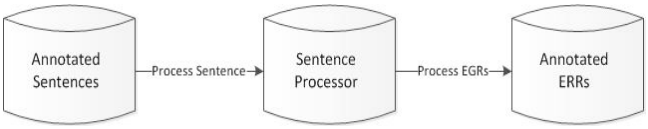

Fig. 3. Offline phase

The first task in the sentence processor is to use a POS tagger to determine the POS tag for each word in the input sentence. The Stanford POS tagger [22] was used for this task. Below is a list of the POS labels that were used.

- PRP: Personal pronoun, e.g. I, he, she, it, they...
- NN: Noun, e.g. Car, ball, computer…
- JJ: Adjective, e.g. separable, nice, great…
- IN: Preposition or subordinating conjunction, e.g. among, below, upon, next…
- VB: Verb, e.g. love, act, say, break…
- RB: Adverb, e.g. occasionally, swiftly, slowly…
- RP: Particle, e.g. apart, aside, across, under…
- WP: Wh-pronoun, e.g. what, which, who…
- CC: Coordinating conjunction, e.g. and, or, but, either…

After the tagging process, we apply the Stanford dependency parser [31], which extracts the dependency tree of the input sentence. We then remove non-emotional content from the tree by applying a set of rules; the goal of applying these rules is to capture the emotional part of the sentence. The rules are categorized into: separation rules and deletion rules.

1. *Separation Rule 1*: Ignore the sentence before the word "but"; same rule applies to words that have the same effect as "but". Since the word "but" implies opposition, the sentence after "but" replaces the emotions present in the first sentence. Figure 4 shows the dependency tree of the following sentence "*it was a bit complicated but we had fun*".

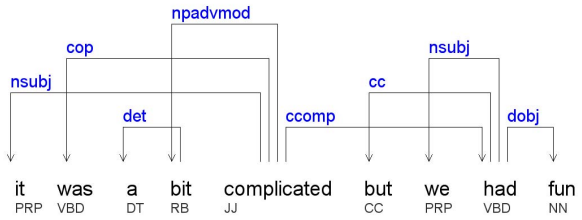

Fig. 4. Dependency Tree of "*it was a bit complicated but we had fun*"

After ignoring the sentence before "but" we get the minimized tree in Figure 5.

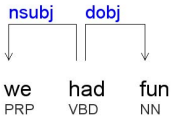

Fig. 5. Minimized Tree after applying eparation Rule 1 to the tree in Fig. 4

2. *Separation Rule 2:* Ignore the sentence after the word "as", if it is followed by a pronoun; same rule applies to words that have the same effect as "as". The word "as" is a subordinate conjunction, which means that the sentence after "as" is a subordinate sentence. The subordinate sentence can be considered as a complement to the meaning of the sentence, and thus can be ignored. For example, the sentence "*people stare as I run*" will be considered as two parts "people stare" and "as I run". However, the second sentence will be ignored.

3. *Deletion Rule 1*: Remove a verb if it has no object and it is connected to a WRB or WP pronoun, as it can be considered as a complement to the emotional meaning of the sentence. Consider for instance the following sentence and its corresponding parse tree in Figure 6: "*where you are going is a disgusting place*".

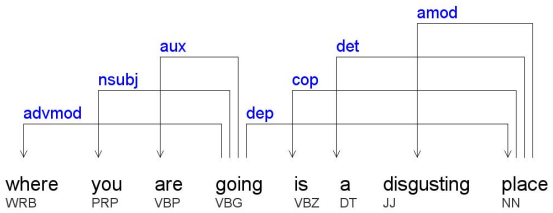

Fig. 6. Dependency Tree of "*where you are going is a disgusting place*"

The part "where you are going" will be removed from the dependency tree, which will keep the part "disgusting place" in the tree to be considered as ERR for the sentence as shown in Figure 7.

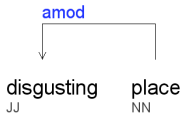

Fig. 7. Minimized Tree after applying Deletion Rule 1 to the tree in Fig. 6

4. *Deletion Rule 2*: Remove a verb node if it is either non-emotional or it is one of the 'to be' verbs because it can also be considered as a complement to the emotional meaning of the sentence. To decide whether a verb is emotional or not, WordNet-Affect [11], SentiWordNet [21] and the emotional probability of this verb in the training set are used. We consider a verb to be emotional if it either exists in WordNet-Affect, has a polarity in SentiWordNet or its emotional probability (extracted from the training set) is above a certain threshold. For example, the previous sentence "we had fun" will be minimized to hold only two nodes "we" and "fun".

5. *Deletion Rule 3:* Remove pronoun nodes if they are not connected to other nodes. For example, if this rule is applied on the previous tree for the sentence "*it was a bit complicated but we had fun*", the only node left is "fun" which will be used as the ERR for the sentence.

B. Examples

In what follows, we give some examples to highlight the inner working and the effectiveness of the above-discussed rules in extracting the emotional parts of a given sentence. Take for instance the sentence "*She makes me happy*". We

first generate its corresponding dependency tree (Figure 8). We then apply the rules presented in the previous section, specifically, Deletion Rule 2 and Deletion Rule 3. Consequently, the dependency tree is pruned resulting in the tree in Figure 9 that contains only the emotional parts of the original sentence. The resulting tree and the sentence's emotion (Happy) is considered an ERR in our model.

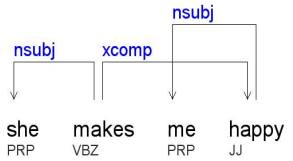

Fig. 8. Dependency Tree of "*She makes me happy"*

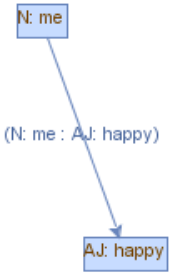

Fig. 9. ERR extracted from the sentence "*She makes me happy"*

Consider another sentence: "*So many lies about who you're talking to, where you're going, what you're doing*". The dependency tree contains many complement sentences as shown in Figure 10.

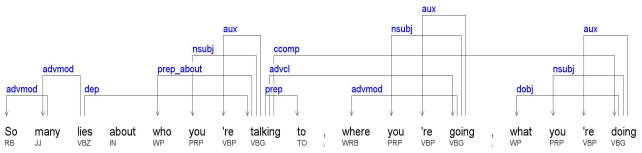

Fig. 10. Dependency Tree for "*So many lies about who you're talking to, where you're going, what you're doing"*

Applying Deletion Rule 1 results in the compact tree (Figure 11) that represents the emotional part of the sentence: "So many lies". The resulting tree and its corresponding emotion (Disgust) represent an ERR in our model.

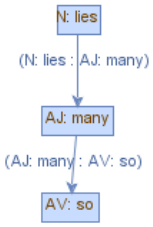

Fig. 11. ERR extracted from the dependency tree in Fig. 10

C. *Comparison and Classification*

The goal of this phase is to compare two ERRs, one that represents an input sentence and one that represents a sentence in the training set. Figure 12 shows a high level overview of our system.

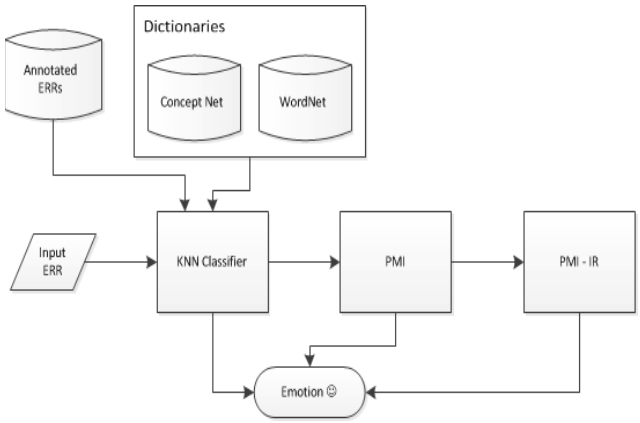

Fig. 12. Classifier Overview

In Figure 12, the Annotated ERRs are the ERRs that were created during the offline phase as described in Section IV-A. For a given input sentence, the system generates its corresponding ERR using the same process as in the offline phase. The generated ERR is compared using a customized KNN algorithm to every ERR in the Annotated ERR dataset. The emotion of the input sentence is assigned to be the emotion of the annotated ERR that achieved the maximum similarity score with the corresponding ERR (i.e., the emotion of the closest neighbor using a 1NN classifier). The intuition here is to find a reference ERR that is similar in structure and meaning to the input ERR. The KNN Classifier makes use of WordNet and ConceptNet for generalization and comparison reasons (more details later). Finally, the emotion is either classified, or PMI followed by PMI-IR are used to classify the emotion.

To measure the similarity between a reference ERR and an input ERR, we built a KNN classifier based on two handcrafted measurements: semantic similarity and keyword similarity. The semantic similarity indicates how much the two ERRs are related in meaning, while the keyword similarity indicates the number of matched words between the two ERRs. The matched ERR is the one that has the maximum semantic similarity. Ties are broken based on the keyword similarity (details are discussed later in this section).

However, using only a KNN classifier might not be sufficient if the training set is small. Hence, we extended our classification process by using other classifiers in case the input ERR was rejected by the KNN classifier. An input ERR is rejected by the KNN classifier if there are no matches, that is, the similarity score to all the training ERRs is equal to zero. If the input ERR was rejected by the KNN classifier, we try to classify it using a PMI Classifier. A PMI classifier reflects how much two words are related to each

other according to a certain dataset. And finally if it was rejected by PMI, we try to classify it using PMI-IR. PMI-IR is a variation of PMI that depends on retrieving information from the World Wide Web. Next, we describe our three classifiers in details.

1) KNN Classifier

To compute the similarity between two ERRs, we use the following similarity function:

Similarity = Sim(VerbNounClauses) + Sim(NounClauses) + Sim(AdjectiveClauses) + Sim(AdverbClauses)

We compute the similarity for each type of construct individually by comparing common synonyms using WordNet. The similarity score is increased by 1 for every match. We also obtain from ConceptNet the similarity of the words' concepts. The obtained value (a number between 0 and 1) is added to the similarity score. In the KNN classifier, we distinguish between two types of similarities: keyword similarity and semantic similarity. The *Sim* function will return a numerical score along with the type of similarity; either keyword or semantic. However, if one of the *Sim* functions returns a semantic similarity score then the overall similarity will be considered as semantic similarity, otherwise it would be treated as a keyword similarity.

In what follows, we explain the details of the KNN classifier. We start first with a brief overview of how WordNet and ConceptNet are used.

a) WordNet

We compare individual words using WordNet to check if they are synonyms or not. If two words are matched, we increase the similarity score by 1.

b) ConceptNet

Comparing concepts is different than comparing words. For example the concept "good time" is related to several other concepts, e.g. "great time", "having a blast" and "Amazing". However each one has a different structure. Thus we defined structural patterns to compare concepts. After comparing concepts, we increase the similarity score by the similarity obtained from ConceptNet. To compare concepts, we used pre-defined concept structures, for instance:

- JJ NN = JJ? NN

The previous rule indicates that an adjective followed by a noun can match either a single noun or a noun proceeded by an adjective. The "?" matches the preceding word 0 or 1 times. For example, "great time" and "fun". Other structures include:

- VB NN = JJ NN? | VB NN
- NN = JJ | NN | JJ NN

However, following these rules may not result in a good concept similarity score, for example, the similarity score between "great time" and "blast" is as low as 0.4. However the similarity score between "good time" and "blast" is 0.72 even though the words "great" and "good" are similar in this context. To solve this issue, we retrieved the top-10 most similar terms to the word "great". For each similar term (of the same POS tag) $term^i$ that is above a certain threshold, we compute the similarity between "$term^i$ time" and "blast". As a final concept similarity score, we pick the maximum value of all the similarities as follows:

Score = Max{ ConceptNetSim("$term^i$ time*", "*blast*") }*

The same procedure is done whenever two ERRs are being compared. Having introduced WordNet and ConceptNet we now discuss how the comparison is made.

c) Comparing VNCs:

We compare each input VNC with all reference VNCs and pick the max match (the one that generates max similarity). To compare VNCs together we used the following procedure:

- First compare if the verbs are matched (either synonyms or belong to related concepts)
- If the verbs are matched (similar in meaning), then we compare the nouns of each VNC; comparison is done by checking WordNet for synonyms and ConceptNet for concepts.
- If any two nouns are matched, we check their adjectives and adverbs in a similar way.

d) Comparing Nouns

We consider an ERR to be a set of nouns if it has only nouns or it contains only non-emotional verbs (a verb is non-emotional if it is not in the list of emotional verbs collected from WordNet-Affect). According to this definition we distinguish between two cases:

- If both, the reference and the input ERRs are represented as set of Nouns, we compare all nouns in both ERRs. If two nouns are matched, we also check the similarity of their verbs, if they exist, which will affect the similarity score of those nouns.
- If only the input ERR can be treated as a set of nouns, then we compare all nouns of the input ERR with only nouns from the reference ERR.

e) Comparing adjectives/adverbs

We compare the adjectives and adverbs list of both ERRs.

In each step while comparing concepts, we perform generalization using WordNet and ConceptNet i.e. we expand the word to its synonyms and related concepts.

Finally after computing the similarity score between two ERRs, the classifier needs to decide whether it is Semantic or

Keyword similarity. The similarity is considered semantic similarity if one of the following conditions is correct:

- Emotional VNC matched (an emotional verb, its nouns, adjectives and adverbs were matched). We used WordNet-Affect to get a list of emotional verbs. For example, the sentence "I love her" is not similar to the sentence "I love home". However the sentence "I love her" is similar to "I love kids".
- All nouns are matched and both ERRs can be treated as a set of nouns (two nouns are only matched if they both have a similar set of adjectives and adverbs).
- Adjectives/Adverbs are matched and both ERRs have no other components (No VNCs or NNs)
- All words in both ERRs are matched (same sentence). The input ERR contains the reference ERR.

If any of the previous conditions was true, the similarity between the two ERRs is considered a semantic similarity otherwise it will be considered a keyword similarity.

2) Comparison Example using KNN

Now we give a simple example to clarify the aforementioned process. Consider the sentences "*she looks gorgeous*" and "*she is beautiful*". The ERRs generated from both sentences are shown in Figure 13.

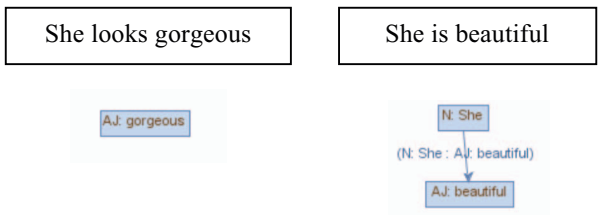

Fig. 13. Comparison two ERRs

When comparing the two ERRs, we start by comparing both adjectives "beautiful" and "gorgeous". Using WordNet, we discover that these words are synonyms. Since the tree of the sentence "she is beautiful" contains the tree of the sentence "she looks gorgeous" it means the similarity is a semantic similarity and both sentences hold the same meaning. If the two words were not synonyms, we would have checked whether the two adjectives have the same concept using ConceptNet. Therefore, the similarity score of these two sentences is 1 and it is a semantic similarity since one of the semantic similarity conditions is true.

3) PMI & PMI-IR

As a backup classification method, we used PMI and PMI-IR. PMI is a measure of association between two random variables. It quantifies the discrepancy between the probability of their coincidence given their joint distribution and their individual distributions. We start by computing the emotional probability for each emotional label in our dataset by using the following equation:

PMI(E; w) = log (#Sentences with emotion E that contain the word w / #total sentences that contain w)

For an input sentence we look for the emotional probability of each word in the sentence, and choose the word with the maximum emotional probability to be the cause of the emotion. If all the emotional probabilities were under a certain threshold, we use PMI-IR, a variation of PMI where the number of hits retrieved from a major search engine (in our experiments Google) is used to compute the emotional probability. Multiple queries for a sentence will be sent to the search engine. We start by sending the terms extracted from the ERR without including any emotions, then for each emotion, we add the emotion to the search query. Finally, we compute the PMI using the following formula:

PMI(E; s) = log (#Number of hits retrieved for the query (s + E) / #Number of hits retrieved for the query (s))

V. Experiments

Our system follows a 2-tier client-server architecture. The client is responsible for getting the input sentence to be classified from the user (or the dataset to be classified), and sending it to the server for processing. On the server side, the generalization on the data is made and results are computed and sent back to the client.

To setup our experiments, WordNet and SentiWordNet dictionaries, and Stanford NLP tools were installed on a machine running Windows 7, 64-bits with 4 GB of RAM. We also installed ConceptNet on a server machine, Dell T7500, with 48 GB of RAM and multi-core Intel® Xeon® processors. The reason behind installing ConceptNet on the server machine is because ConceptNet needs to upload many JSON files to a local SOLR sever (full-text search server), which creates a large index that does not fit on a normal machine's memory. ConceptNet computations are thus done on the server side and results are sent back to the client.

We tested our method using two different training sets. In the first experiment, the training and testing sets are both from the same dataset. While in the second experiment the training set is totally different from the testing set.

1) Experiment I

We tested our classifiers first on Aman dataset [26]; we took sentences with length between 0 and 10 words to be our training set of ERRs. Our assumption is that short sentences contain emotional concepts (not emotional words), for example the sentence "*thank you my friends*" contains one simple emotional concept. On the other hand, a long sentence might contain several misleading emotional concepts that are not related to the emotion of the sentence, e.g., "*As of late, it seems that everything my parents do, they do it just to annoy me, they're great parents*". In this sentence there are two emotional concepts "they annoy me" and "great parents". However, the emotion of the sentence is anger, causing the emotional concept "great parents" to be recognized as anger concept, which is not accurate. That is why we relied on short sentences for training. Based on this assumption, we built our training data from 570 sentences.

We selected our testing data to be sentences with length between 10 and 15 words. The results are reported in Table II under the "ERR-Based" column, for each of the 6 Ekman emotions.

The "Baseline" column shows the results of a naïve baseline that recognizes emotions by checking the presence of one or more emotional words in the sentence [30]. It counts the number of emotional words of each category in the sentence, and then assigns this sentence the emotions with the largest number of words.

TABLE II. EXPERIMENT I RESULTS

Emotion	Precision		Recall		F-score	
	ERR-Based	Baseline	ERR-Based	Recall	ERR-Based	F-score
Happiness	0.91	0.589	0.94	0.390	0.92	0.469
Sadness	0.86	0.527	0.83	0.283	0.84	0.368
Disgust	0.80	0.944	0.78	0.099	0.79	0.179
Anger	0.85	0.681	0.78	0.262	0.82	0.379
Surprise	0.81	0.318	0.75	0.296	0.78	0.306
Fear	0.93	0.824	0.85	0.365	0.89	0.506

As shown in Table II, our system (ERR-Based) achieves better results than the baseline and a high precision in classifying all the six emotions where the highest precision achieved is 93% for the emotion fear. The average F-score of our system is 84% for all of the six emotions.

We also compared our results to the EmoHeart classifier results reported in [19] on the same testing set (Aman dataset). As shown in Figure 14, our system achieved a better F-Score for each of the 6 emotions.

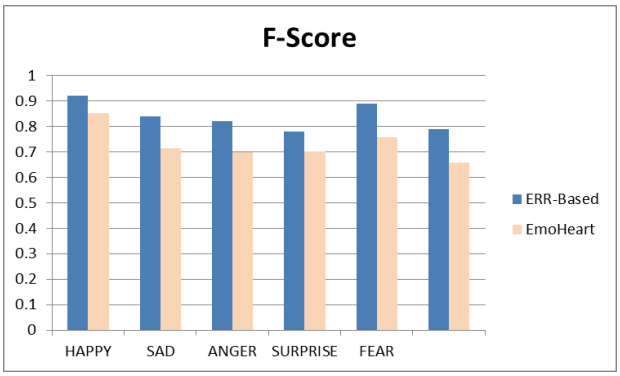

Fig. 14. Comparison to EmoHeart on Aman dataset on all six emotions

2) Experiment II

In this experiment, we used a set of sentences collected from twitter, where each sentence has as a set of hash tags related to it [28]. Each collected tweet was automatically labeled with one emotion according to its emotion hash tag. For example, the emotion of the sentence is considered to be "anger" if one of the anger hash tags was found in the sentence (i.e. #irritation, #annoyance, #irritating). As a result, a dataset of annotated sentences was generated. A total of 2,488,982 tweets were collected. We randomly selected 18,000 sentences to be our training set. However, the training dataset we used had some drawbacks. First, it did not cover the "Disgust" emotion i.e. one of the 6 Ekman emotions was not present. As a result, we decided to exclude the "Disgust" emotion from the testing set, as it has no reference sentences in the training set. Second, many of the sentences had grammatical mistakes, as they were written in informal English. For example, the sentence "*if it hurts it hurts, that's no*" contains meaningless part ("that's no"). We removed those sentences from our training dataset when spotted. We tested our classifier using the tweet training set we collected and compared our results to EmoHeart classifier [19] on the same testing set (Aman dataset). The results are shown in Figure 15.

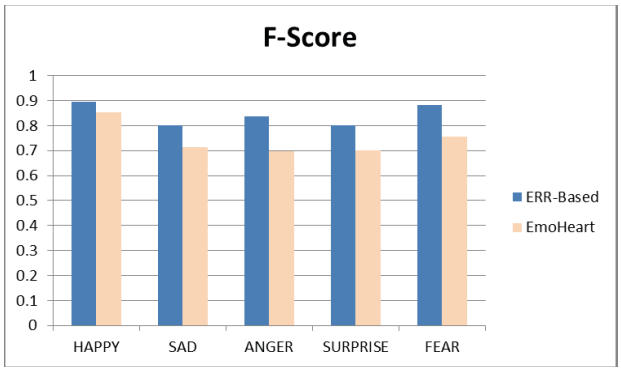

Fig. 15. Comparison to EmoHeart on Aman dataset

As shown in the results, our system achieved a better F-score than EmoHeart when classifying the emotions. In fact, the improvement is almost 10%. The highest precision achieved in this experiment was 89% for the emotion Happy and an average F-score of 84% for the five emotions. By having a large training set that covers all the required emotions, we believe, based on the results above, that we could have an emotion classifier from text that can classify any English sentence. For future work, we plan to build an interactive website that accepts sentences from users and gives back the corresponding annotation on the fly. All data and code will be made public.

VI. CONCLUSION

In this work, we introduced a new approach for classifying emotions from textual data based on a fine-grained level. Our contribution lies in performing complex syntactic and semantic analysis of the sentence and using various ontologies such as Wordnet and ConceptNet in the process of emotion recognition. Syntactic and semantic analysis of the sentence makes our classifier context sensitive, while using Wordent and ConceptNet helps our classifier generalize the training set, which leads to better coverage of emotion rules.

We evaluated our approach on two different datasets, one consisting of blog posts and one consisting of tweets. Our approach outperformed the state-of-the-art method in emotion classification from text (EmoHeart). We showed that comparing the relations between the words of the sentence could lead to better accuracy than assigning an

individual emotional rate for each word. We also showed that even with a training set different from the test set, our classifier performs better than EmoHeart. Moreover, the architecture of the proposed classifier is very flexible allowing it to be easily extended to classifying any number of emotions by providing a reasonably-sized training set that covers the required emotions.

REFERENCES

[1] Zahra Khalili, and Mohammad Hasan Moradi. "Emotion recognition system using brain and peripheral signals: using correlation dimension to improve the results of EEG." *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009.

[2] Jeffrey F. Cohn, and Gary S. Katz. "Bimodal expression of emotion by face and voice." *Proceedings of the sixth ACM international conference on Multimedia: Face/gesture recognition and their applications*. ACM, 1998.

[3] Liyanage C. De Silva, and Pei Chi Ng. "Bimodal emotion recognition." *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000.

[4] Torao Yanaru, "An emotion processing system based on fuzzy inference and subjective observations." *Artificial Neural Networks and Expert Systems, 1995. Proceedings., Second New Zealand International Two-Stream Conference on*. IEEE, 1995.

[5] EC-C. Kao, et al. "Towards Text-based Emotion Detection A Survey and Possible Improvements." *Information Management and Engineering, 2009. ICIME'09. International Conference on*. IEEE, 2009.

[6] Erik Cambria, Andrew Livingstone, and Amir Hussain. "The hourglass of emotions." *Cognitive behavioural systems*. Springer Berlin Heidelberg, 2012. 144-157.

[7] P. Ekman, (1999) Basic emotions. In T. Dalgleish and T. Power (Eds.) *The handbook of cognition and emotion*. Pp. 45-60. New York.: John Wiley & Sons.

[8] Jeffrey T. Hancock , Christopher Landrigan, and Courtney Silver. "Expressing emotion in text-based communication." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007.

[9] Charles E. Osgood. *Cross-cultural universals of affective meaning*. University of Illinois Press, 1975.

[10] Charles E. Osgood, and Oliver Tzeng. *Language, meaning, and culture: The selected papers of CE Osgood*. Praeger Publishers, 1990.

[11] Carlo Strapparava, and Alessandro Valitutti. "WordNet Affect: an Affective Extension of WordNet." *LREC*. Vol. 4. 2004.

[12] Carlo Strapparava, and Rada Mihalcea. "Learning to identify emotions in text." *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008.

[13] Radim Burget, Jan Karasek, and Zdeněk Smekal. "Recognition of emotions in Czech newspaper headlines." *Radioengineering* 20.1 (2011): 39-47.

[14] Dung T. Ho, and Tru H. Cao. "A high-order hidden Markov model for emotion detection from textual data." *Knowledge Management and Acquisition for Intelligent Systems*. Springer Berlin Heidelberg, 2012. 94-105.

[15] Klaus R. Scherer, and Harald G. Wallbott. "Evidence for universality and cultural variation of differential emotion response patterning." *Journal of personality and social psychology* 66.2 (1994): 310.

[16] Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. "Emotion recognition from text using semantic labels and separable mixture models." *ACM transactions on Asian language information processing (TALIP)* 5.2 (2006): 165-183.

[17] Cheng-Yu Lu, et al. "Automatic event-level textual emotion sensing using mutual action histogram between entities." *Expert systems with applications* 37.2 (2010): 1643-1653.

[18] François-Régis Chaumartin. "UPAR7: A knowledge-based system for headline sentiment tagging." *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 2007.

[19] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. "EmoHeart: conveying emotions in second life based on affect sensing from text." *Advances in Human-Computer Interaction* 2010 (2010): 1.

[20] George A. Miller, "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.

[21] Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." *Proceedings of LREC*. Vol. 6. 2006.

[22] K. Toutanova, Klein D., Manning C., Singer Y., StanfordPOStagger, [Online].Available:http://nlp.stanford.edu/software/tagger.shtml,Stanford, 2003

[23] Hui Yang, et al. "A hybrid model for automatic emotion recognition in suicide notes." *Biomedical informatics insights* 5.Suppl 1 (2012): 17.

[24] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. "Hierarchical versus flat classification of emotions in text." *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, 2010.

[25] Second Life Game, http://secondlife.com/

[26] Saima Aman, and Stan Szpakowicz. "Identifying expressions of emotion in text." *Text, Speech and Dialogue*. Springer Berlin Heidelberg, 2007.

[27] Erik Cambria, Haixun Wang, and Bebo White. "Guest Editorial: Big Social Data Analysis." *Knowledge-Based Systems 69: 1-2* (2014).

[28] Wenbo Wang, et al. "Harnessing twitter" big data" for automatic emotion identification." *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 2012.

[29] Colin Humphries, et al. "Syntactic and semantic modulation of neural activity during auditory sentence comprehension." *Journal of cognitive neuroscience* 18.4 (2006): 665-679.

[30] Saima Aman, and Stan Szpakowicz. "Using Roget's Thesaurus for Fine-grained Emotion Recognition." *IJCNLP*. 2008.

[31] Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. "Generating typed dependency parses from phrase structure parses." *Proceedings of LREC*. Vol. 6. 2006.