DeepEmo: Learning and Enriching Pattern-Based Emotion Representations

Elvis Saravia
National Tsing Hua University
Hsinchu, Taiwan
ellfae@gmail.com

Hsien-Chi Toby Liu
National Tsing Hua University
Hsinchu, Taiwan
tobboymailbox@gmail.com

Yi-Shin Chen
National Tsing Hua University
Hsinchu, Taiwan
yishin@gmail.com

Abstract

We propose a graph-based mechanism to extract rich-emotion bearing patterns, which fosters a deeper analysis of online emotional expressions, from a corpus. The patterns are then enriched with word embeddings and evaluated through several emotion recognition tasks. Moreover, we conduct analysis on the emotion-oriented patterns to demonstrate its applicability and to explore its properties. Our experimental results demonstrate that the proposed techniques outperform most state-of-the-art emotion recognition techniques.

1 Introduction

Emotions can be defined as conscious affect attitudes, which constitute the display of a feeling. An emotion classification task consists of the representation learning or manual feature extraction of emotional words and phrases. Although there is constant debate about what exactly constitutes an emotion (Weidman et al., 2017), there is no doubt of the societal and economic benefits that emotion recognition models and their applications can offer. Emotions are key influencers to understand other human social behaviors, such as motivation, interest, sarcasm, and mental health. Recently, emotion detection capabilities have been embedded into empathy-aware, AI conversational agents, such as Woebot \(^1\) and in the dialogue system proposed by (Zhou et al., 2017). The motivation of our work stems from the need to better model and explore different forms of online emotional expressions, particularly implicit ones. The proposed emotion representations allows emotion recognition systems to consider linguistic components such as stop words, which are usually ignored in emotion analysis but form an integral part of how we express our emotions and opinions (Pennebaker et al., 2007).

Emotion recognition from text is challenging since emotional expressions can be highly implicit and are subject to evolve over time. This presents a challenge when relying on resources (e.g., emotion lexicons) that were generated by hand-crafted linguistic rules. For instance, mispronounced words appearing together will not be identified as the same when applying conventional feature extractors such as bag of words and n-grams. Another common tendency in online social networks is the use of different forms of expression, such as slang, code words and emoticons, to express feelings and opinions. To address this problem, we design an algorithm, based on graph-theory, similar to (Santos et al., 2017), to automate the process of extracting emotion representations.

As an overview, we first collect an emotional corpus through noisy labels, which is then modeled via distant supervision as in (Go et al., 2009). Then, emotion features are extracted via a graph-based mechanism, which are further enriched with word embeddings in order to preserve semantic meaning between patterns. To evaluate the quality of patterns, emotion detection models are trained using various online classifiers and deep learning models. Our main contributions are summarized as follows: 1) A graph-based mechanism for automatic emotion-based feature extraction, 2) a set of emotion-rich feature representations used to conduct various emotion recognition tasks and other relevant target tasks, 3) a comprehensive performance analysis of various conventional learning models and deep learning models as it applies to emotion recognition from text, and 4) an emotion-rich lexicon, which is offered as open source, that allows for deeper analysis of a given emotion-relevant corpus.

\(^1\)https://woebot.io/
2 Related Work

2.1 Overview of Feature Representations
We compare various feature extractors against the proposed technique, across two dimensions: 1) Coverage - the features should be able to capture important implicit and explicit emotional information, and 2) Adaptability - the features can apply to other type of emotional corpora, originating from different domains. Recent emotion recognition systems employ representation learning for feature detection (Poria et al., 2016; Savigny and Purwarianti, 2017; Nguyen and Nguyen, 2017; Abdul-Mageed and Ungar, 2017). In general, a combination of word embeddings (e.g., word2vec (Mikolov et al., 2013)) as input and a deep learning model, such as convolutional neural network (CNN), performs well for sentence classification (Kim, 2014; Zhang et al., 2015; Felbo et al., 2017). Due to the nature of these type of models and the type of features they learn, they tend to have high coverage, high adaptability, require little supervision (i.e., features are automatically learned), and capture context to some extent. However, there is a well-known trade-off between interpretability and high performance with these type of models. Our graph-based feature extraction mechanism focuses more on the underlying interaction between linguistic components. Therefore, the patterns automatically surface both implicit and explicit emotional expressions.

2.2 Emotion Corpus and Models
There are several open affective datasets, such as SemEval-2007 Affective Text Task (Strapparava and Mihalcea, 2007) and Olympic games dataset (Sintsova et al., 2013). However, these emotion datasets are either limited by lack of fine-grained emotion labels or quantity. We bootstrap a set of noisy labels used to obtain larger collections of emotional tweets, and then perform annotations via distant supervision similar to (Read, 2005; Go et al., 2009; Mintz et al., 2009; González-Ibáñez et al., 2011; Mohammad, 2012; Purver and Battersby, 2012; Wang et al., 2012; Mohammad and Kiritchenko, 2015; Abdul-Mageed and Ungar, 2017). In emotion recognition studies, the Plutchik’s wheel of emotions (Plutchik, 2001) or Ekman’s six basic emotions (Ekman, 1992), are commonly adopted to define emotion categories (Mohammad, 2012; Suttles and Ide, 2013). Emoticons and emojis have also proven to be useful for defining emotion categories (Eisner et al., 2016; Felbo et al., 2017). Similar to (Mohammad and Kiritchenko, 2015; Liew and Turtle, 2016; Abdul-Mageed and Ungar, 2017), we rely on hashtags to define our emotion categories.

2.3 Emotion Lexica
Emotion classifiers have enabled understanding of mood patterns displayed by mental health patients (Park et al., 2012; De Choudhury et al., 2013; Harman and Dredze, 2014; Coppersmith et al., 2014). Some of these studies rely on a pre-defined lexicon, such as LIWC (Pennebaker et al., 2007), WordNet Affect (Strapparava et al., 2004) and EmoLex (Mohammad and Turney, 2013), to extract emotional cues from text-based corpora. A recent study demonstrates the correlation between emotional tone and perceived demographic traits among users in a social network (Volkova and Bachrach, 2016). This study relies on an emotion detection system, which is built using lexical features, such as emoticons and hashtags (Pang et al., 2002). Other user information, such as age and gender were obtained from external sources, which limited the amount of data that the authors could collect. An improvement to their work would be to use the content from the users’ tweets to automatically determine user attributes, such as age and gender (Sap et al., 2014). Other works use hand-crafted linguistic features to improve emotion classification performance (Blitzer et al., 2007; Wang et al., 2012; Roberts et al., 2012; Qadir and Riloff, 2013; Volkova et al., 2013; Mohammad and Kiritchenko, 2015; Volkova and Bachrach, 2016; Becker et al., 2017). These features are useful for emotion classification but offer limited coverage. Our emotion lexicon is constructed with an emphasis on coverage (i.e., captures implicit and explicit emotional expressions).

3 Methodology

3.1 Graph-Based Representations
In this section, we introduce a graph-based feature extraction algorithm, which automatically extracts a set of emotion-rich syntactic patterns. For notation purposes, we denote scalars with italics (e.g., $u$), vectors with bold lowercase (e.g., $v$), and matrices with bold uppercase (e.g., $X$). The patterns $P = \{p_1, p_2, ..., p_n\}$ will be assigned a weight, also referred to as a pattern score, which is used

---

2 LIWC stands for linguistic inquiry and word count.
to determine how important a pattern $p$ is to an emotion $e$. In the context of an emotion classifier, patterns and their weights play the role of features. The graph-based feature extraction algorithm is summarized in the following steps:

**Step 1 (Normalization):** First, two separate collection of documents – subjective tweets $S$ (obtained through hashtags as noisy labels) and objective tweets $O$ (obtained from news accounts) – are obtained using the Twitter API. Both datasets are tokenized by white-spaces and then further preprocessed by applying lower case and replacing user mentions and URLs with a `<usermention>` and `<url>` placeholder, respectively. Hashtags, which is achieved in two steps: 

- Vertices $V$ is a set of nodes which represent the tokens extracted from the corpus. Edges, denoted as $A$, represent the relationship of words as extracted from a piece of text using a window approach. This consideration is important as it preserves the prosody and underlying syntactic structure of textual data. For instance, a post “<usermention> last night’s concert was just awesome !!!!! <hashtag>” results in the following set of arcs: “<usermention> → last”, “last → night”, ... , “!!!!!! → <hashtag>”.

**Step 2 (Graph Construction):** Given the normalized objective tweets $O$ and subjective tweets $S$, two graphs are constructed: objective graph $G_o(V_o; A_o)$ and subjective graph $G_s(V_s; A_s)$, respectively. Vertices $V$ is a set of nodes which represent the tokens extracted from the corpus. Edges, denoted as $A$, represent the relationship of words as extracted from a piece of text using a window approach. This consideration is important as it preserves the prosody and underlying syntactic structure of textual data. For instance, a post “<usermention> last night’s concert was just awesome !!!!! <hashtag>” results in the following set of arcs: “<usermention> → last”, “last → night”, ... , “!!!!!! → <hashtag>”.

**Step 3 (Graph Aggregation):** The goal of this step is to obtain a set of arcs that are more relevant to subjectivity or emotional expressions. The assumption is that by adjusting graph $G_s$ with $G_o$, it is possible to obtain a new graph $G_e$, also referred to as an emotion graph. $G_e$ preserves emotion-relevant tokens, which is achieved in two steps:

1. For an arc $a_i \in A$, its normalized weight can be computed as shown in Equation 1.

\[
w(a_i) = \frac{freq(a_i)}{\max_{j \in A} freq(a_j)}
\]

where $freq(a_i)$ is the frequency of arc $a_i$.

2. Subsequently, new weights for arcs $a_i \in G_e$ are assigned based on a pairwise adjustment as shown in Equation 2.

\[
w(a_i) = \begin{cases} 
w(a_i) - w(a_o), & \text{if } a_o = a_i \in G_o \\
w(a_o), & \text{otherwise} \end{cases}
\]

The resulting weights belonging to graph $G_e$ were adjusted so that the most frequently occurring arcs in objective set $G_o$ are weakened in $G_e$. As a result, arcs in $G_e$ that have higher weights represent tokens that are more relevant to subjective content. Furthermore, arcs $a_i \in A_e$ are pruned based on a threshold $\phi_w$.

**Step 4 (Token Categorization):** Given an adjacency matrix $M$, an entry $M_{i,j}$ is computed as:

\[
M_{i,j} = \begin{cases} 
1 & \text{if node } i \text{ and } j \text{ are linked in } G_e \\
0 & \text{otherwise} \end{cases}
\]

Then, eigenvector centrality and clustering coefficient of all vertices in $V_e$ are computed, which will be used to categorize tokens into two types: *connector words* and *subject words*.

1. **Connector Words:** To measure the influence of all nodes in graph $G_e$, we utilize eigenvector centrality, which is computed as:

\[
c_i = \frac{1}{\lambda} \sum_{j \in V_e} M_{i,j} c_j
\]

where $\lambda$ denotes a proportionality factor and $c_i$ is the centrality score of node $i$.

Given $\lambda$ as the corresponding eigenvalue, Equation 4 can be reformulated in vector notation form as $M c = \lambda c$, where $c$ is an eigenvector of $M$. Given a selected eigenvector $c$ and the eigenvector centrality score of node $i$, denoted as $c_i$, the final list of connected words, hereinafter referred to as $CW$, is obtained by retaining all tokens with $c_i > \phi_{eig}$. $CW$ represents the set of words that are very frequent and contain high centrality (e.g., “or”, “and”, and “my”).

2. **Subject Words** In contrast, subject words or topical words are usually clustered together, i.e., many subject words are interconnected by the same connector words. Therefore, a coefficient is

---

3. Each dataset contains over 2+ million tweets.

4. $\phi_w$ is an experimentally defined threshold.

5. $\phi_{eig}$ is an experimentally defined threshold.
assigned to all nodes in $G_e$ and is computed as:

$$c_l = \frac{\sum_{j \neq i; k \neq j, k \neq i} M_{i,j} \times M_{i,k} \times M_{j,k}}{\sum_{j \neq i; k \neq j, k \neq i} M_{i,j} \times M_{i,k}} \times \frac{1}{|V_e|}$$

where $c_l$ denotes the average clustering coefficient of node $i$, which captures the amount of inter-connectivity among neighbours of node $i$. Similar to the connector words, the subject words, hereinafter referred to as $SW$, are obtained by retaining all the tokens with $c_l > \phi_d$. 6 Examples of subject words are (e.g., “never” and “life”).

**Step 5 (Pattern Candidates):** Given the set of tokens, $SW$ and $CW$, we employ a bootstrap approach to construct candidate patterns which express subjective meaning without losing syntactic structure. Consequently, the following are some of the rules which are used to define the candidate patterns: $<sw,sw,cw>$, $<cw,sw>$, and $<cw,cw,sw>$, where $sw$ and $cw$ represent arbitrary tokens obtained from the set $SW$ and $CW$, respectively. It’s important to clarify that sequences of size two and three were used in this work since this setting experimentally worked best for us. We may sometimes refer to these candidate patterns as templates, similar to (Riloff, 1996; Riloff and Wiebe, 2003; Tromp and Pechenizkiy, 2015). The difference in our work is that we don’t impose grammatical heuristics or rules in the pattern extraction process, therefore, our patterns tend to naturally have higher coverage and capture both implicit and explicit emotional content.

**Step 6 (Basic Pattern Extraction):** A naive pattern extraction process consists of applying the syntactic templates to a training corpus in an exhaustive manner. In addition, subject words $sw$ in each pattern is replaced with a $<*>$ placeholder. This operation allows for unknown subject words, not present in our training corpus, to be considered when modeling on an external emotional corpus. We are interested in patterns that are highly associated with subjectivity, so patterns frequently occurring above a threshold are kept and the rest are filtered out. 8 In Table 1, we provide examples of the type of basic patterns extracted along with the corresponding templates. Next, we discuss the process of enriching the syntactic patterns with word embeddings. This enrichment process helps to preserve semantic between patterns and improves feature relevance (Santos et al., 2017).

| Templates                | Pattern Examples |
|--------------------------|------------------|
| $<cw,sw>$                | “stupid *”, “like *”, “am *” |
| $<cw,cw,sw>$             | “love you *”, “shut up *” |
| $<cw,cw,su>$             | “wo for *” |
| $<sw,cw,cw>$             | “* on the” , |
| $<sw,cw,su>$             | “* <hashtag>” |

Table 1: Examples of patterns and templates extracted through the basic pattern extraction mechanism.

### 3.2 Enriched Patterns

#### Weighted Word Embeddings

First, we obtain pre-trained Twitter-based word embeddings from (Deriu et al., 2017) and reweigh them via a sentiment corpus through distant supervision (Read, 2005; Go et al., 2009). 9 We trained a fully connected deep neural network with 10 epochs (1 hidden layer) via backpropagation as in (Deriu et al., 2017). We denote the sentiment word embeddings as $W \in \mathbb{R}^{d \times n}$, where $d = 52$. Note that term frequency-inverse document frequency ($tf-idf$) was used to reduce the vocabulary of words (from 140K to 20K words).

#### Word Clusters

We then apply agglomerative clustering to generate clusters of semantically related words through their word embedding information. To determine the quality of the clusters they are compared with WordNet-Affect synsets (Strapparava et al., 2004) and tested for both homogeneity and completeness. We use Ward’s method (Ward Jr, 1963) as the linkage criterion and cosine distance as the distance metric. In the end, we obtained $k = 1500$ clusters. We use the scikit-learn implementation to perform the word clustering (http://scikit-learn.org).

#### Enriched-Pattern Construction

The purpose of the word clusters is to use them to guide the process of enriching the patterns. In other words, the patterns will hold some semantic relationship, which becomes useful for classification problems. Note that this process is similar to the naive pattern extraction with the exception of the word embedding integration. This entails a bootstrap process where an emotional corpus is processed and candidate patterns are searched in an exhaustive fashion. Any word sequences in the emotional corpus

---

6 $\phi_d$ is an experimentally define threshold.
7 Subjective dataset $S$ is used to process the templates.
8 A grand total of 19,821 patterns were extracted.
that satisfies the templates are retained and the rest are filtered out. In addition, the \(sw\) component of the templates must be a word found in the word clusters defined above. Furthermore, patterns that appear < 10 are filtered out, producing a total of 187,647 patterns. In Section 6, we analyze the patterns more in depth and provide examples.

3.3 Emotion Pattern Weighing

The patterns extracted in the previous step are still not mapped to any specific emotion category. Before training a classification model, a pattern weighing mechanism needs to be employed. Similar to other popular weighing mechanisms, such as \(tf-idf\), weights determine the importance of patterns to each emotion \(e_j \in E\). The proposed pattern weighing scheme is a modification of \(tf-idf\), coined as pattern frequency-inverse emotion frequency \((pf-ief)\), and is defined in two steps. Firstly, we compute for \(pf\) as:

\[
pf_{p,e} = \log \frac{\text{freq}(p, e) + 1}{\text{freq}(p, e)} + 1
\]

where \(\text{freq}(p, e)\) represents the frequency of \(p\) in \(e\), and \(pf_{p,e}\) denotes the logarithmically scaled frequency of a pattern \(p\) in a collection of texts related to emotion \(e\).

Then we compute for \(ief\) as:

\[
ief_p = \log \frac{\text{freq}(p, e) + 1}{\sum_{e_j \in E} \text{freq}(p, e_j) + 1}
\]

where the inverse emotion frequency \(rief_p\) is a measure of the relevance of pattern \(p\) across all emotion categories.

Finally, we obtain a pattern score as:

\[
ps_{p,e} = pf_{p,e} \times ief_p
\]

where \(ps_{p,e}\) is the final score that reflects how important a pattern \(p\) is to an emotion class \(e\).

4 Models

4.1 DeepEmo

The proposed framework, coined as DeepEmo, combines a multilayer-layer CNN architecture with a matrix form of the proposed graph-based features. The input \(X \in \mathbb{R}^{n \times m}\) denotes an embedding matrix where entry \(X_{i,j}\) represents the pattern score of enriched pattern \(i\) in emotion \(j\).\(^{10}\)

The input is fed into 2 1d convolutional layers with filters of size 3 and 16. The output of this process is passed through a non-linear activation function (i.e., ReLU (Nair and Hinton, 2010)) and produces a feature map matrix. A 1-max pooling layer (Boureau et al., 2010) of size 3 is then applied to each feature map. The results of the pooling are fed into two hidden layers of dimensions 512 and 128 in that order, each applied a dropout (Hinton et al., 2012) of 0.8 for regularization. We chose a batch size of 128 and trained for 7 epochs using Adam (Kingma and Ba, 2014) optimizer. A softmax function is used to generate the final classification. We use Keras (Chollet et al., 2015) to implement the CNN architecture.

4.2 Vector Model

As a baseline, we present a naive vector model (EVM), which demonstrates basic usability and applicability of the basic patterns proposed in Section 3.1. Pattern weights are obtained using the pattern weighing mechanism proposed in Section 3.3. Formally, given \(n\) patterns and \(m\) emotions, we can represent the entire emotion model as matrix \(EM \in \mathbb{R}^{n \times m}\). An entry \(E_{i,j}\) represents the rank of basic pattern \(i\) in emotion \(j\), which is based on the pattern score \(ps_{i,j}\). Note that patterns with higher \(ps\) values have lower rank values, as in they are more relevant to that particular emotion. Assume a social post \(tw\) for which we want to obtain its portrayed emotion, we first compute its frequency vector \(f \in \mathbb{R}^n\), where entry \(f_i\) represents the frequency of pattern \(i\) in input social post \(d\). We compute the emotion scores as:

\[
es = f \cdot EM
\]

where \(es \in \mathbb{R}^m\) and entry \(es_j\) corresponds to the final emotion score of emotion \(j\) for the post \(tw\). The index of the minimum of these values is selected as the final emotion detected for \(tw\).

4.3 Comparison Models

4.3.1 Traditional models

We compare DeepEmo against various traditional methods (e.g., bag of words (BoW), character-level (char), n-grams, TF-IDF) commonly used in sentence classification. The classifier used to train these models is the stochastic gradient descent (SGD) classifier provided by scikit-learn.
4.3.2 Deep Learning models

Deep learning architectures enable automatic learning of features from textual information. We observed that among the works that employ deep learning models for emotion classification, they vary by the choice of input: pretrained word/character embeddings and end-to-end learned word/character representations. Our work differs in that we utilize enriched graph-based representations as input, therefore, we believe it is also important to compare with these methods. We compare with convolutional neural networks (CNNs), recurrent neural networks (RNNs), bidirectional gated recurrent neural networks (GRNNs), and word embeddings (word2vec) (Mikolov et al., 2013).

| Models   | Features             | anger | anticipation | disgust | fear | joy | sadness | surprise | trust | F1 Avg. |
|----------|----------------------|-------|---------------|---------|------|-----|---------|----------|-------|---------|
| BoW      | word frequency       | 0.53  | 0.08          | 0.17    | 0.53 | 0.71| 0.60    | 0.36     | 0.33  | 0.57    |
| BoW(TF-IDF) | TF-IDF              | 0.55  | 0.09          | 0.18    | 0.57 | 0.73| 0.62    | 0.39     | 0.35  | 0.60    |
| n-gram  | word frequency       | 0.56  | 0.09          | 0.17    | 0.57 | 0.73| 0.64    | 0.42     | 0.39  | 0.61    |
| n-gram(TF-IDF) | TF-IDF              | 0.58  | 0.12          | 0.17    | 0.60 | 0.75| 0.67    | 0.47     | 0.45  | 0.63    |
| char    | character frequency  | 0.35  | 0.03          | 0.04    | 0.20 | 0.51| 0.46    | 0.10     | 0.12  | 0.37    |
| char(TF-IDF) | TF-IDF              | 0.33  | 0.03          | 0.06    | 0.21 | 0.52| 0.45    | 0.11     | 0.13  | 0.37    |
| char_ngram | character frequency | 0.49  | 0.06          | 0.12    | 0.46 | 0.67| 0.59    | 0.30     | 0.28  | 0.52    |
| char_ngram(TF-IDF) | TF-IDF              | 0.53  | 0.07          | 0.15    | 0.53 | 0.71| 0.59    | 0.35     | 0.31  | 0.57    |
| word2vec | word embeddings     | 0.50  | 0.02          | 0.13    | 0.48 | 0.69| 0.51    | 0.35     | 0.31  | 0.53    |
| LIWC     | affect words         | 0.35  | 0.03          | 0.11    | 0.30 | 0.49| 0.35    | 0.18     | 0.19  | 0.35    |
| EVM      | patterns             | 0.24  | 0.02          | 0.04    | 0.38 | 0.50| 0.34    | 0.20     | 0.21  | 0.38    |
| CNN-patt | basic patterns       | 0.47  | 0.00          | 0.00    | 0.45 | 0.67| 0.61    | 0.15     | 0.08  | 0.52    |
| DeepEmo  | enriched patterns    | 0.58  | 0.16          | 0.32    | 0.65 | 0.75| 0.70    | 0.59     | 0.55  | 0.67    |

Table 2: Comparison of our model against conventional feature extractors using F1-score. LIWC uses a bag of words approach. word2vec model adopts pre-trained embeddings from (Mikolov et al., 2013). char refers to character-level features. n-gram employ unigrams, bigrams, and trigrams as features. CNN-patt uses the proposed CNN architecture with basic patterns.

5 Experiments

5.1 Data

We follow (Mohammad, 2012; Wang et al., 2012; Abdul-Mageed and Ungar, 2017) and construct a set of hashtags (grounded on Plutchik’s wheel of emotions (Plutchik, 2001)) to collect English tweets from Twitter API. Specifically, we use the eight basic emotions of Plutchik: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The hashtags serve as noisy labels, which allows annotation of the data through distant supervision (Go et al., 2009). 339 hashtags were defined in total. To ensure tweets quality, we follow pre-processing steps proposed by (Abdul-Mageed and Ungar, 2017) and considered the hashtag appearing in the last position of a tweet as the ground truth. We split the data into training (90%) and testing (10%). The final distribution of the data and a list of hashtag examples for each emotion are provided in Table 3. In the following sections, we evaluate the effectiveness of the enriched patterns on several emotion recognition tasks. We use F1-score as the evaluation metric, which is commonly used in emotion recognition studies due to the imbalanced nature of the emotion datasets.

5.2 Experimental Results

Traditional Feature Extractors The results obtained from the traditional feature extractors are presented in Table 2. As the table shows, TF-IDF models usually produce better results than basic count-based features for both character-level and word-level feature extractors. These findings are consistent with the work of (Zhang et al., 2015), where traditional methods, such as n-gram TF-IDF, were found to perform comparable to neural networks on various sentence classification tasks.

Results with Pattern Approaches The results of EVM and CNN-patt, which employ the basic graph-based patterns, are worst that most of the conventional approaches. DeepEmo, which uses the enriched patterns, acquires better results (F1-score of 67%) than both CNN-patt and EVM, and all of the other conventional approaches. In fact, our method obtains the best F1-score on all emotions. We can also observe that there is a significant boost in performance (+15%) when using the
enriched patterns (DeepEmo) as compared to the basic patterns model (CNN-patt). Overall, we can observe that the enriched graph-based features are feasible for training emotion recognition models.

Comparison to state-of-the-art: We also compare results with published literature, which employ emotion recognition systems using Ekman’s six basic emotions. For fair comparison, we reduced our dataset from eight emotions to six emotions: anger, disgust, fear, joy, sadness, and surprise. As shown in Table 5, our emotion recognition system achieves better results (F1-score of 0.72%) than most of the methods with the exception of (Volkova and Bachrach, 2016). Their emotion recognition system performs better than ours (F1-score of 78%) since they use well-defined linguistic features, such as emoticons and hashtags. Our features are more susceptible to noise because we aim for higher coverage in order to capture more implicit emotional expressions. This consideration is important if we intend to use the emotion lexicons for conducting deep analysis on affective datasets. In addition, their features are domain-specific, which means some important features, such as emoticons and hashtags, may not be applicable to other affective datasets. According to (Zhang et al., 2015), traditional methods are strong candidates on this type of tasks for dataset of size up to the hundreds of thousands, and only after several millions do CNN models start to do better. We plan to continue enlarging our datasets and refining pattern weights, which are feasible methods to improve results.

6 Analysis of Enriched Patterns

In this section, we explore the enriched patterns extracted from a gender-based dataset. We collected user feeds from Twitter and classified users into male and female classes based on their content via Sap et al.’s gender predictor (Sap et al., 2014). This produces a gender dataset, which we also manually verify by ourselves. We randomly sampled 2000 males and 2000 females and then randomly sampled 100 tweets from each user feed. This generated 400,000 tweets in total, which we further reduced by filtering out tweets with \( \leq 5 \) words. The final amount of tweets is 294,792, which we classify using DeepEmo.

We apply a pattern frequency analysis on the gender data using the enriched patterns. The patterns that are shared by both males and females are discarded and the 1000 most frequently occurring patterns for each gender dataset are analyzed. Examples of the most frequent emotional patterns captured by \( < cw, sw > \) and \( < sw, cw > \) templates as expressed by both females and males are provided in Table 7. The words inside the {} represent the subject words captured by the pattern.
enrichment process. We can observe that subject words represent emotion-rich words such as “despise”, “yelling”, and “loneliest”. The connecting words, on the other hand, provide context, which helps to better understand the enriched patterns.

We are currently investigating whether there are gender-specific emotional patterns or expressions on social media. However, it is too early to derive conclusions from the primitive analysis presented here. We can still observe that providing context helps to tell a story behind the emotional expressions. Another interesting research direction would be to use the patterns directly for gender prediction. The goal of the analysis was to explore the enriched patterns and show how they may be used for conducting deeper analysis on an emotional corpus.

**Pattern Coverage** We computed the coverage of the enriched patterns on several affective datasets. As shown in Table 6, 89.4% of the tweets in the gender data contains at least one of the enriched patterns. Our patterns also show high coverage on datasets from different domains, such as SST-2 (76%), SST-5 (71%) (Socher et al., 2013), and PsychExp (95%) (Wallbott and Scherer, 1988). We observed that the dataset size did not influence the coverage results. A high coverage (95%) was obtained on emotional experiences described in (Wallbott and Scherer, 1988), which originate from a different domain from which the patterns were constructed. This shows that our enriched patterns are adaptable to other domains, which open opportunities for further exploration and experimentation.

### Table 6: Statistics on word coverage per text of the enriched patterns on several affective datasets. * denotes that we used the El-oc testing data. The numbers inside the () represent the number of classes present in the dataset.

| Emotion Dataset   | Study          | Task          | Domain          | Dataset Size | Enriched Patterns |
|-------------------|----------------|---------------|-----------------|--------------|-------------------|
| Our Full Dataset  | Ours           | Emotion (8)   | Tweets          | 1,896,849    | 0.94              |
| Gender Data       | Ours           | Emotion (8)   | Tweets          | 294,792      | 0.89              |
| SemEval07 Task 14 | (Strapparava and Mihalcea, 2007) | Emotion (3)   | Headlines       | 801          | 0.62              |
| SemEval17 Task 4  | (Rosenthal et al., 2017) | Sentiment (3) | Tweets          | 20,021       | 0.99              |
| SemEval18 Task 1  | (Mohammad and Bravo-Marquez, 2017) | Emotions (4) | Tweets          | 3590         | 0.92              |
| SST-2             | (Socher et al., 2013) | Sentiment (5) | Reviews         | 58,990       | 0.76              |
| SST-5             | (Socher et al., 2013) | Sentiment (5) | Reviews         | 96,660       | 0.71              |
| PsychExp          | (Wallbott and Scherer, 1988) | Emotion (3)   | Experiences     | 7339         | 0.95              |

### Table 7: Examples of the top 1000 most frequently occurring patterns by gender.

| Emotions     | Male patterns                                                                 | Female patterns                                           |
|--------------|-------------------------------------------------------------------------------|-----------------------------------------------------------|
| Anger        | 2[crazy ; you](despise ; like)[try]                                          | my [yelling ; would][want ; hate][you]                    |
| Sadness      | your [lyrics ; bouncing ; your]                                              | better [come ; you](wreck ; despise)[going]              |
| Surprise     | last [second] ; to [announce]                                                | happy [birthday ; only][person]                           |
| Fear         | {you} have ; [getting ; your]                                                 | my [stepmom ; the][loneliest]                            |

7 Discussion

Abdul-Mageed and Ungar (2017) showed that improving data quality is an important step in improving emotion classification results (achieves an F1-score of 83%). We observed that they report a larger dataset (790,059) and more balanced data collections for each emotion. In contrast, our dataset is more imbalanced, but even when balancing the results did not improve significantly (average F1-score of 68%). At the time of writing this manuscript, the authors were still working on making their datasets publicly available, so unfortunately we couldn’t compare directly with their method. As future work, we hope to keep refining our hashtags and improving the emotional corpus. All benchmark datasets, lexicons, pre-trained models, and code for running the models will be made available soon.

8 Conclusion

We proposed an enriched graph-based feature extraction mechanism to extract emotion-rich representations. The patterns are enriched with word embeddings and are used to train several effective emotion recognition models. Our patterns capture implicit emotional expressions which improves emotion recognition results and helps with interpretability. We demonstrate a basic application of the proposed affective lexicon on a gender dataset. We hope to improve the pattern weighing mechanism so as to improve the performance on emotion recognition tasks and minimize trade-off between pattern coverage and performance.
References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 718–728.

Karin Becker, Viviane P Moreira, and Aline GL dos Santos. 2017. Multilingual emotion classification using supervised learning: Comparative experiments. Information Processing & Management, 53(3):684–704.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, box-office and blenders: Domain adaptation for sentiment classification. In Proceedings of the 45th annual meeting of the association of computational linguistics, pages 440–447.

Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th international conference on machine learning (ICML-10), pages 111–118.

François Chollet et al. 2015. Keras. https://github.com/keras-team/keras.

Glen Coppersmith, Mark Dredze, and Craig Harman. 2014. Quantifying mental health signals in twitter. In Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pages 51–60.

Munmun De Choudhury, Scott Counts, and Eric Horvitz. 2013. Predicting postpartum changes in emotion and behavior via social media. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 3267–3276. ACM.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multilanguage sentiment classification. In Proceedings of the 26th International Conference on World Wide Web, pages 1045–1052. International World Wide Web Conferences Steering Committee.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. arXiv preprint arXiv:1609.08359.

Paul Ekman. 1992. An argument for basic emotions. Cognition & emotion, 6(3-4):169–200.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. arXiv preprint arXiv:1708.00524.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(2009):12.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2, pages 581–586. Association for Computational Linguistics.

GACCT Harman and Mark H Dredze. 2014. Measuring post traumatic stress disorder in twitter. In ICWSM.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Ilya Ivanov. 2017. Sentiment bi-rnn. https://github.com/ilivans/attention-sentiment.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.3882.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Jasy Suet Yan Liew and Howard R Turtle. 2016. Exploring fine-grained emotion detection in tweets. In Proceedings of the NAACL Student Research Workshop, pages 73–80.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1003–1011. Association for Computational Linguistics.

Saif M Mohammad. 2012. Emotional tweets. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 246–255. Association for Computational Linguistics.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. arXiv preprint arXiv:1708.03696.
Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Saif M Mohammad and Peter D Turney. 2013. Crowd-sourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Huy-Thanh Nguyen and Minh-Le Nguyen. 2017. Sentence modeling with deep neural architecture using lexicon and character attention mechanism for sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 536–544.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Minsu Park, Chiyoung Cha, and Meeyoung Cha. 2012. Depressive moods of users portrayed in twitter. In *Proceedings of the ACM SIGKDD Workshop on healthcare informatics (HI-KDD)*, volume 2012, pages 1–8. ACM New York, NY.

James W Pennebaker, Roger J Booth, and Martha E Francis. 2007. Linguistic inquiry and word count: Liwc [computer software]. Austin, TX: liwc.net.

Robert Plutchik. 2001. The nature of emotions human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350.

Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. 2016. Convolutional mkl based multimodal emotion recognition and sentiment analysis. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 439–448. IEEE.

Matthew Purver and Stuart Battersby. 2012. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics.

Ashequl Qadir and Ellen Riloff. 2013. Bootstrapped learning of emotion hashtags# hashtags#you. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 2–11.

Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop*, pages 43–48. Association for Computational Linguistics.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics.

Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sandra M Harabagiu. 2012. Empatweet: Annotating and detecting emotions on twitter. In *LREC*, volume 12, pages 3806–3813.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation. SemEval ’17*, Vancouver, Canada. Association for Computational Linguistics.

Leandro B dos Santos, Edilson A Corrêa Jr, Osvaldo N Oliveira Jr, Diego R Amancio, Letícia L Mansur, and Sandra M Aluísio. 2017. Enriching complex networks with word embeddings for detecting mild cognitive impairment from speech transcripts. *arXiv preprint arXiv:1704.08088*.

Maarten Sap, Gregory Park, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Lyle Ungar, and Hansen Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1146–1151.

Julio Savigny and Ayu Purwarianti. 2017. Emotion classification on youtube comments using word embedding. In *Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), 2017 International Conference on*, pages 1–5. IEEE.

Valentina Sintsova, Claudiu-Cristian Musat, and Pearl Pu. 2013. Fine-grained emotion recognition in olympic tweets based on human computation. In *4th Workshop on computational approaches to subjectivity, sentiment and social media analysis*, EPFL-CONF-197185.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In Proceedings of the 4th International Workshop on Semantic Evaluations, pages 70–74. Association for Computational Linguistics.

Carlo Strapparava, Alessandro Valitutti, et al. 2004. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086.

Jared Suttles and Nancy Ide. 2013. Distant supervision for emotion classification with discrete binary values. In International Conference on Intelligent Text Processing and Computational Linguistics, pages 121–136. Springer.

Erik Tromp and Mykola Pechenizkiy. 2015. Pattern-based emotion classification on social media. In Advances in Social Media Analysis, pages 1–20. Springer.

Svitlana Volkova and Yoram Bachrach. 2016. Inferring perceived demographics from user emotional tone and user-environment emotional contrast. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 1567–1578.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 505–510.

Harald G Wallbott and Klaus R Scherer. 1988. How universal and specific is emotional experience?: Evidence from 27 countries on five continents.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing twitter” big data” for automatic emotion identification. In Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), pages 587–592. IEEE.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Aaron C Weidman, Conor M Steckler, and Jessica L Tracy. 2017. The jingle and jangle of emotion assessment: Imprecise measurement, casual scale usage, and conceptual fuzziness in emotion research. *Emotion*, 17(2):267.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2017. Emotional chatting machine: emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*.

HUSEIN ZOLKEPlI. 2017. Emotion classification. https://github.com/huseinzol05/Emotion-Classification-Comparison.