# CARER: Contextualized Affect Representations for Emotion Recognition

**Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, Yi-Shin Chen***

National Tsing Hua University
Hsinchu, Taiwan
{ellfae, tobbymailbox, yenhao0218, phejimlin, yishin} @gmail.com

## Abstract

Emotions are expressed in nuanced ways, which varies by collective or individual experiences, knowledge, and beliefs. Therefore, to understand emotion, as conveyed through text, a robust mechanism capable of capturing and modeling different linguistic nuances and phenomena is needed. We propose a semi-supervised, graph-based algorithm to produce rich structural descriptors which serve as the building blocks for constructing contextualized affect representations from text. The pattern-based representations are further enriched with word embeddings and evaluated through several emotion recognition tasks. Our experimental results demonstrate that the proposed method outperforms state-of-the-art techniques on emotion recognition tasks.

## 1 Introduction

Emotions reflect different users' perspectives towards actions and events, therefore they are innately expressed in dynamic linguistic forms. Capturing these linguistic variations is challenging because it involves knowledge of linguistic phenomena such as slang and coded words. Previous methods model these linguistic behaviours through rule-based (Volkova and Bachrach, 2016) and statistics-based approaches (Becker et al., 2017). These methods construct features that rely on hand-crafted resources; thus, they cannot properly capture the *evolving* linguistic variability found in large-scale opinionated content.

Consider the social posts *"Thanks God for everything"* and *"Tnx mom for waaaaking me two hours early. Cant get asleep now"*, a lexicon-based model may not properly represent the emotion-relevant phrases: *"waaaaking me"*, *"Thanks God"*, and *"Tnx mom"*. First, the word *"waaaaking"* doesn't exist in the English vocabulary, hence its referent may vary from its standard form, *"waking"*. Secondly, knowledge of the semantic similarity between the words *"Thanks"* and *"Tnx"* is needed to establish any relationship between the last two phrases. Even if such relationship can be established through knowledge-based techniques, it's difficult to reliably determine the association of these phrases to a group of emotions. This is because traditional methods analyze data at the sentence level, which may be less effective as compared to methods that model the corpus as a complex network (Santos et al., 2017).

We represent an emotion corpus as a graph, which may suffer less from the problems mentioned above. This method efficiently captures the global mutual use of linguistic variations found in textual information. This is particularly important for linguistic behaviour that is socially and culturally influenced, as is common in opinionated content. Other advantages of the graph approach are that minimum domain knowledge and manual effort are required to capture important *contextual* and *latent* information, which are useful to disambiguate meaning in emotional expressions.

As an overview, we first collect an emotion dataset through noisy labels, annotated via *distant supervision* as in (Go et al., 2009). The graph-based mechanism helps to construct contextualized, pattern-based emotion features, which are further enriched with word embeddings in order to preserve semantic relationship 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 as follows: 1) A graph-based algorithm for automatic emotion-relevant feature extraction, 2) a set of emotion-rich feature representations enhanced through word embeddings, 3) and a comprehensive performance analysis of various con-

* Corresponding author



*Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697
Brussels, Belgium, October 31 - November 4, 2018. 

ventional learning models and deep learning models used for text-based emotion recognition.

The rest of the paper is organized as follows: Section 2 discusses the relevant literature and different aspects of emotion recognition research addressed in this work; then, Section 3 provides details of the proposed methodology for extracting contextualized emotion-relevant representations; next, Section 4 lists the constructed emotion recognition models and comparison models; later, Section 5 discusses the data collection and experimental results; and finally, Section 6 further explains key insights observed from the results.

## 2 Related Work

**Emotion Lexica:** Several works use hand-crafted features and statistics-based approaches to train emotion recognition models (Blitzer et al., 2007; Wang et al., 2012; Roberts et al., 2012; Qadir and Riloff, 2013; Volkova et al., 2013; Becker et al., 2017; Saravia et al., 2016a). Some of these studies rely on affect lexicons, such as *LIWC* (Pennebaker et al., 2007) and *WordNet Affect* (Strapparava et al., 2004), to extract emotion features from a text-based corpus. A recent study trained emotion detection systems built on emoticons and hashtag features (Volkova and Bachrach, 2016). Hand-crafted features are useful for emotion recognition but are usually constrained by manually created resources. Our graph-based features are obtained in an semi-supervised manner, requiring minimum domain expertise and no dependency of linguistic resources that quickly become outdated.

**Emotion Corpora:** There are several affective datasets such as *SemEval-2017 Task 4* (Rosenthal et al., 2017) and *Olympic games* dataset (Sintsova et al., 2013). However, these datasets are limited by quantity. We bootstrap a set of noisy labels to obtain large-scale emotion tweets, and then perform annotation via distant supervision as in (Go et al., 2009; González-Ibánez et al., 2011; Wang et al., 2012; Mohammad and Kiritchenko, 2015; Abdul-Mageed and Ungar, 2017). In emotion recognition studies, 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). Similar to previous works, we rely on hashtags to define our emotion categories.

**Feature Representations:** Recent emotion recognition systems employ *representation learning* for automatic feature extraction (Poria et al., 2016; Savigny and Purwarianti, 2017; Abdul-Mageed and Ungar, 2017). In general, a combination of word embeddings (Mikolov et al., 2013) and a convolutional neural network (CNN) performs well for sentence classification tasks (Kim, 2014; Zhang et al., 2015). These models learn features which tend to have high coverage, high adaptability, require minimum supervision, and capture contextual information to some extent. We aim to leverage them and combine them with the proposed affect representations. Our graph-based feature extraction algorithm focuses on the underlying interactions between important linguistic components. Graph analysis measurements then help to output the building blocks for constructing pattern-based features. Hence, the patterns can be constructed to capture important contextual and latent emotion-relevant information.

## 3 Contextualized Affect Representations

In this section, we introduce a graph-based algorithm which helps to output the building blocks used to bootstrap a set of emotion-rich representations. The structural descriptions offered by the graph are particularly efficient at automatically surfacing important information (i.e., contextual and latent information) from a large-scale emotion corpus. Two different measurements are used to surface two families of words, which are in turn used to construct contextualized, pattern-based affect representations. The patterns are further enriched using word embeddings so as to preserve semantic relationship between patterns. After the patterns are constructed, the goal is to assign a weight to each pattern, referred to as a *pattern score*, which denotes how important a pattern $p$ is to an emotion $e$. In the context of emotion classification, 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, we collected two separate datasets using the Twitter API: subjective tweets $S$ (obtained through hashtags as weak labels) and objective tweets $O$ (obtained from Twitter feeds of news accounts).[1] Both datasets are tokenized by white-spaces and then preprocessed by applying lower case and replacing user mentions and URLs with a <*usermention*> and <*url*>

[1]Each dataset contains 2+ million tweets. $S$ was collected using 339 hashtags, similar to the process in Section 5.1.

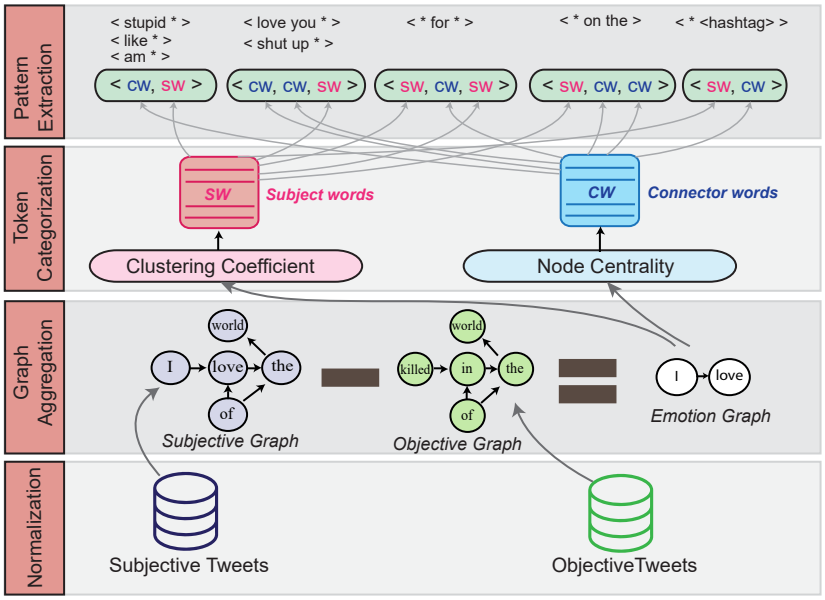


Figure 1: An overview of the important steps used to generate graph-based pattern representations.

placeholder, respectively. Hashtags are used as ground-truth in this work, so to avoid any bias we replace them with a *<hashtag>* placeholder.

**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 extracted using a window approach. These steps help to preserve the syntactic structure of the data. Given a post *"<usermention> last night's concert was just awesome !!!!! <hashtag>"*, the resulting arcs are: *"<usermention> → last"*, *"last → night"*, ... , *"!!!!! → <hashtag>"*.

**Step 3 (Graph Aggregation):** In this step we obtain a set of arcs that represent syntactic structures more common in subjective content. By adjusting graph $G_s$ with $G_o$, we obtain a graph $G_e$, referred to as the *emotion graph*, which preserves emotion-relevant tokens and is obtained in two steps:

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

$$w(a_i) = \frac{freq(a_i)}{\max_{j \in A} freq(a_j)} \quad (1)$$

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.

$$w(a_i) = \begin{cases} w(a_{s_i}) - w(a_{o_j}), & \text{if } a_{o_j} = a_{s_i} \in G_o \\ w(a_{s_i}), & \text{otherwise} \end{cases} \quad (2)$$

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 common in subjective content. Furthermore, arcs $a_i \in A_e$ are pruned based on a threshold $\phi_w$[2].

**Step 4 (Token Categorization):** Two different graph measurements are used to extract two family of words from $G_e$. These will function as the building blocks to build contextualized patterns. We formalize this step as follows: Given an adjacency matrix $\mathbf{M}$, an entry $M_{i,j}$ is computed as:

$$M_{i,j} = \begin{cases} 1 & \text{if node } i \text{ and } j \text{ are linked in } G_e \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Then, the eigenvector centrality and clustering coefficient of all vertices in $V_e$ are computed and used to categorize tokens into two types:

(1) **Connector Words:** To measure the influence of all nodes in graph $G_e$, we utilize eigenvec-

[2] $\phi_w$ is an experimentally determined threshold.

tor centrality, which is calculated as:

$$c_i = \frac{1}{\lambda} \sum_{j \in V_e} M_{i,j} c_j \quad (4)$$

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 $\mathbf{Mc} = \lambda \mathbf{c}$, where $\mathbf{c}$ is an eigenvector of $\mathbf{M}$. Given a selected eigenvector $\mathbf{c}$ and the eigenvector centrality score of node $i$, denoted as $c_i$, the final list of connector words, hereinafter referred to as $CW$, is obtained by retaining all tokens with $c_i > \phi_{eig}$ [3]. $CW$ correspond to the set of nodes that are very frequent and have high connectivity to high-rank nodes (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 assigned to all nodes in $G_e$ and is computed as:

$$cl_i = \frac{\sum_{j \neq i; k \neq j; k \neq i} M_{i,j} \times M_{i,k} \times M_{j,k}}{\sum_{j \neq i; k \neq j; k \neq i} M_{i,j} \times M_{i,k}} \times \frac{1}{|V_e|} \quad (5)$$

where $cl_i$ denotes the average clustering coefficient of node $i$ which captures the amount of interconnectivity 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 $cl_i > \phi_{cl}$ [4]. Examples of subjects words obtained are: *"never"* and *"life"*.

The subject words represent psychological oriented words similar to the LIWC affect lexicon (Pennebaker et al., 2007), while connector words reflect the set of most common words in the subjective tweets (e.g., pronouns, auxiliary verbs, and conjunctions). As presented by Chung and Pennebaker (2007), both connector words and subject words are important for conveying emotion. Influenced by their work, we aim to capture intricate relationships – through the graph – between these two families of words. The graph structure helps to preserve syntax and can automatically be used to surface emotion-relevant information.

One of the advantages of using graphs to represent syntactic relationships is that rare and important words are also surfaced. As shown in Table 1, informal words and misspellings, such as *"definetley"*, *"happnd"*, were surfaced. Words containing character repetitions help to express emotion intensity (e.g., *"plzzzzzzz"*, *"aaaaaaah"*, and *"yayyyyy"*). Interestingly, emotion-related coded words are also captured (e.g., *"juju"*, *"sh*t"*, *"4ever"*, and *"baobei"*[5]). All these examples show the benefit of using graph methods to capture emotion-relevant linguistic information.

| Subject Words (SW) | Connector Words (CW) |
|---|---|
| baobei, juju | myy, ??!, urs, congrats |
| plzzzzzzz, aaaaaaah | bcoz, jus, tsk |
| happnd, yayyyyy | sh*t, smh, smhh, pfft |
| definetley, everytin | 4ever, stfu, eff |

Table 1: Examples of subject words and connector words automatically extracted from the emotion graph $G_e$.

**Step 5 (Pattern Candidates):** Given $SW$ and $CW$, we bootstrap candidate patterns, which are more prevalent in opinionated content, while preserving syntactic structure. We provide the templates used to define the candidate patterns in Table 2. ($sw$ and $cw$ represent arbitrary tokens obtained from the sets $SW$ and $CW$, respectively). It is important to clarify that sequences of size two and three were used in this work since this setting experimentally produced the best results.

**Step 6 (Basic Pattern Extraction):** A naive pattern extraction process consists of applying the syntactic templates to a dataset $S_p$ [6] in an exhaustive manner. In addition, the $sw$ component 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 constructing features. This can enable many useful applications, such as applying the patterns to other domains. 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. In Table 2, we provide examples of the type of basic patterns extracted along with their corresponding templates.

### 3.1 Enriched Patterns

As they stand, the patterns constructed in the previous step contain limited information relevant to emotion classification. Therefore, the patterns are enriched using continuous word representations so as to preserve semantic relationship between pat-

[3] $\phi_{eig}$ is an experimentally determined threshold.

[4] $\phi_{cl}$ is an experimentally defined threshold.

[5] *baobei* is a Chinese word used to show strong affection.

[6] Dataset $S_p$ (size=2+ mil.) is separately collected using similar steps as the subjective dataset $S$, mainly to avoid bias.

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

Table 2: Examples of patterns and templates extracted through the basic pattern extraction mechanism.

terns. The motivation behind this step is to focus on patterns that may be more useful for an emotion classification task. Alternatively, the whole universe of patterns can also be used, but we show in the experiments that the former method significantly improves emotion recognition results.

**Pre-trained Word Embeddings:** First, we obtain Twitter-based, pre-trained word embeddings from (Deriu et al., 2017) and reweight them via a sentiment corpus through distant supervision (Read, 2005; Go et al., 2009).[7] We trained a fully connected neural network (1 hidden layer) with 10 epochs via backpropagation as in (Deriu et al., 2017). The embeddings size is $d = 52$. 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. The scikit-learn package (Pedregosa et al., 2011) was used to compute a total of $k = 1500$ clusters.

**Enriched-Pattern Construction:** The purpose of the word clusters is to enrich the patterns by preserving the semantic relationship between them, which is useful for classification purposes. We achieve this by revising the universe of patterns obtained from the *basic pattern extraction* step, and check to see if the words represented by the $sw$ component exist in any of the word embedding clusters. This is done in an exhaustive manner, ensuring that all possible patterns in the dataset $S_p$ are processed to meet the criteria. Furthermore, patterns that appear $< 10$ times in dataset $S_p$ are filtered out, producing a total of 476,174 patterns.

[7] We collected approximately 10 million tweets via sentiment emoticons (5+ mil. negative and 5+ mil. positive).

The resulting *enriched patterns*[8] now contain both the semantic information provided by the word embeddings and the contextual information provided through the graph components, hence the term *contextualized affect representations*.

### 3.2 Emotion Pattern Weighting

Before using the patterns for classification, they need to be weighted using a weighting mechanism such as *tf-idf* (Leskovec et al., 2014). The weights determine the importance of patterns to each emotion category. The proposed pattern weighting scheme used in this work is a customized version of *tf-idf*, coined as *pattern frequency-inverse emotion frequency* (*pf-ief*), and is defined in two steps. Firstly, we compute for $pf$ as:

$$pf_{p,e} = \log \frac{\left( \sum_{p_i \in P_e} freq(p_i, e) \right) + 1}{freq(p, e) + 1} \quad (6)$$

where $freq(p, e)$ 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:

$$ief_p = \log \frac{freq(p, e) + 1}{\left( \sum_{e_j \in E} freq(p, e_j) \right) + 1} \quad (7)$$

where the inverse emotion frequency $ief_p$ is a measure of the relevance of pattern $p$ across all emotion categories.

Finally, we obtain a pattern score calculated as:

$$ps_{p,e} = pf_{p,e} \times ief_p \quad (8)$$

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

## 4 Models

In this section, we present the emotion recognition models and comparison models used to evaluate the contextualized affect representations. More details are provided in Appendix A.

**CARER:** The proposed framework combines a multi-layer CNN architecture with a matrix form of the *enriched patterns*. The input $\mathbf{X} \in \mathbb{R}^{n \times m}$ denotes an embedding matrix where entry $X_{i,j}$

[8] Refer to Table 6 for enriched patterns examples.

represents the pattern score of enriched pattern $i$ in emotion $j$. [9] $\mathbf{X}$ is fed into two 1-D convolutional layers with filters of sizes 3 and 16. The output of this process is passed through a ReLU activation function (Nair and Hinton, 2010) that 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 0.5 for regularization. We chose a batch size of 128 and trained for 4 epochs using Adam optimizer (Kingma and Ba, 2014). A softmax function is used to generate the final classifications. We use Keras (Chollet et al., 2015) to implement the CNN architecture.

**Baseline Model:** As baseline, we present a first-generation model (**CARER**$_\beta$) that employs primitive *enriched patterns*$^{\ddagger}$[10]. We adopt the CNN architecture used for **CARER**, however, this model differs in that the set of patterns used is significantly smaller as compared to the original size of the enriched patterns. The reason is because a different set of *primitive pattern templates* was used, which captured fewer patterns (187,648). This shows that the proposed method offers flexibility in terms of what templates to use and what size of patterns to generate. This could be useful in cases where there are limited computing and data resources, and for incorporating domain expertise.

**Traditional Models:** We also compare with various traditional methods (bag of words (**BoW**), character-level (**char**), **n-grams**, and **TF-IDF**) which are commonly used in sentence classification. To train the models we use the default stochastic gradient descent (SGD) classifier provided by scikit-learn (Pedregosa et al., 2011).

**Deep Learning Models:** Among the works that employ deep learning models for emotion recognition, they vary by the choice of input: *pre-trained word/character embeddings* and *end-to-end learned word/character representations*. Our work differs in that we utilize enriched graph-based representations as input. We compare with convolutional neural networks (**CNNs**), recurrent neural networks (**RNNs**), bidirectional gated recurrent neural networks (**BiGRNNs**), and word embeddings (**word2vec**) (Mikolov et al., 2013).

[9] We use a zero-padding strategy as in (Kim, 2014).

[10] ‡ hereinafter refers to the primitive enriched patterns.

| Emotions | Amount | Hashtags |
|---|---|---|
| **sadness** | 214,454 | #depressed, #grief |
| **joy** | 167,027 | #fun, #joy |
| **fear** | 102,460 | #fear, #worried |
| **anger** | 102,289 | #mad, #pissed |
| **surprise** | 46,101 | #strange, #surprise |
| **trust** | 19,222 | #hope, #secure |
| **disgust** | 8,934 | #awful, #eww |
| **anticipation** | 3,975 | #pumped, #ready |

Table 3: Data statistics.

## 5 Experiments

### 5.1 Data

We construct a set of hashtags to collect a separate dataset of English tweets from the Twitter API. Specifically, we use the eight basic emotions: *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, and *trust*. The hashtags (339 total) serve as noisy labels, which allow annotation via distant supervision as in (Go et al., 2009). To ensure data quality, we follow the 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%) datasets. The final distribution of the data and a list of hashtag examples for each emotion are provided in Table 3. In the following section 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 Features:** As shown in Table 4, **TF-IDF** models 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**, were found to perform comparable to deep neural networks on various sentence classification tasks.

**Pattern-based Approaches:** The results of **CNN$_{\text{BASIC}}$**[11], which employs the *basic graph-based patterns* proposed in ***Step 6***, perform worse than most of the conventional approaches. Both **CARER**$_\beta$ and **CARER**, which use the *enriched patterns*, acquire better results than **CNN$_{\text{BASIC}}$**

[11] CNN$_{\text{BASIC}}$ adopts CNN architecture of CARER.

and all the other conventional approaches. In fact, our method obtains the best F1-score on all eight emotions. We observed that there are significant gains in performance (↑27% and ↑12%) when using the enriched patterns as compared to the basic patterns and primitive patterns[‡], respectively. This highlights the importance of the pattern enrichment procedure and the benefit of refining the pattern templates. Note that the baseline model, **CARER**$_\beta$, also performs better than all other the comparison models including the state-of-the-art methods (**DeepMoji** and **EmoNet**).

**Comparison to state-of-the-art:** Felbo et al., (2017) proposed a state-of-the-art emotion prediction model, **DeepMoji**, trained on billions of emoji-labeled tweets. We obtained their pre-trained model[12] and applied it to our dataset. As shown in Table 4, their model performs as well as other traditional methods. However, our model (**CARER**) significantly outperforms theirs (↑20%). Moreover, we re-implemented the GRNN model proposed in (Abdul-Mageed and Ungar, 2017). We also outperform their model (**EmoNet**) which manually trains word embeddings, similar to **DeepMoji**. The **CNN$_{w2v}$** model uses word embeddings trained on billions of tweets (Deriu et al., 2017), thus it performs better than all the other approaches, and closer to ours.

**Results with Deep Learning:** We offer more comparison with other various deep learning models as evaluated on Ekman's six basic emotions (i.e., *sadness*, *disgust*, *anger*, *joy*, *surprise*, and *fear*). For the **RNN$_{w2v}$** and **CNN$_{char}$** models, different inputs are used, as shown in Table 5. We feed the enriched patterns as embeddings to a **bidirectional GRNN**, which along with **CARER$_{EK}$** and **CARER**$_\beta$ outperform all the other methods.

**Contextualized Approaches**: **DeepMoji** is built on a stack of Bi-LSTM layers and performs much better with *six* emotion classes. However, using the enriched patterns as input, **CARER$_{EK}$**[13] performs the best (81%). Note that the number of epochs used to train our models is much lower as compared to the other methods, which provides a strong case of the benefit of contextualizing features prior to training the models. Moreover, the important distinction between *connector words* and *subject words* helps to refine and surface relevant contextual information. We also show that the enriched patterns can be applied to other deep learning models besides CNN, such as **BiGRNN**, which leaves an opportunity to explore more complex architectures and fusion models in the future. More importantly, for problems that require deeper understanding of contextualized information, there is a need to go beyond traditional features and distributed representations.

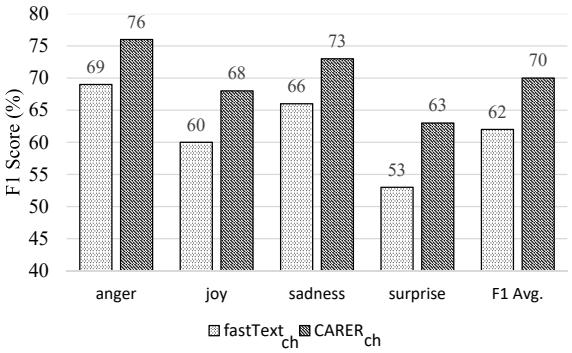


Figure 2: Comparison against Chinese word vectors.

**Multilingual Capabilities:** We also tested the effectiveness of the proposed feature extraction algorithm for the Chinese language. We collected Traditional Chinese datasets[14] from several of Facebook's fan pages and applied the same procedures as were done for the English datasets. User comments are considered as documents and the associated user reaction to the root post represents the emotion labels. For comparison, we obtained Chinese pre-trained word vectors computed through (Bojanowski et al., 2017), and trained a model (**fastText$_{ch}$**) using the proposed CNN architecture. For our approach (**CARER$_{ch}$**), we applied the same CNN architecture on the Chinese-based enriched patterns. As shown in Figure 2, our model performs significantly better on all four emotions (average F1 score of 70%). Overall, we show that the approach is not restricted to any specific language and that the enriched features are applicable to other languages and data sources. In the future, we seek to expand our methods to support other complex languages, such as Japanese, French, and Spanish, where there tends to exist fewer linguistic resources.

## 6 Analysis of Enriched Patterns

### 6.1 Pattern Coverage and Consistency

One of the advantages of the contextualized enriched patterns is that they possess high coverage

[12]Model obtained from github.com/bfelbo/deepmoji

[13]The proposed model trained on six emotions dataset.

[14]Details of the dataset are provided in Appendix B.

| 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$_{\text{TF-IDF}}$** | **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$_{\text{TF-IDF}}$** | **TF-IDF** | 0.58 | 0.12 | 0.17 | 0.60 | **0.75** | 0.67 | 0.47 | 0.45 | 0.63 |
| **char_ngram** | **character frequency** | 0.49 | 0.06 | 0.12 | 0.46 | 0.67 | 0.55 | 0.30 | 0.28 | 0.52 |
| **char_ngram$_{\text{TF-IDF}}$** | **TF-IDF** | 0.53 | 0.07 | 0.15 | 0.53 | 0.71 | 0.59 | 0.35 | 0.31 | 0.57 |
| **LIWC** | **affective words** | 0.35 | 0.03 | 0.11 | 0.30 | 0.49 | 0.35 | 0.18 | 0.19 | 0.35 |
| **CNN$_{\text{w2v}}$** | **word embeddings** | 0.57 | 0.10 | 0.15 | 0.63 | **0.75** | 0.64 | 0.61 | **0.70** | 0.65 |
| **EmoNet** | **word embeddings** | 0.36 | 0.00 | 0.00 | 0.46 | 0.69 | 0.61 | 0.13 | 0.25 | 0.52 |
| **DeepMoji** | **word embeddings** | 0.60 | 0.00 | 0.03 | 0.49 | **0.75** | 0.67 | 0.20 | 0.27 | 0.59 |
| **CNN$_{\text{BASIC}}$** | **basic patterns** | 0.65 | 0.10 | 0.22 | 0.64 | 0.73 | 0.56 | 0.15 | 0.08 | 0.52 |
| **CARER$_{\beta}$** | **enriched patterns**‡ | 0.61 | 0.31 | 0.34 | 0.67 | **0.75** | 0.68 | 0.60 | 0.55 | **0.67** |
| **CARER** | **enriched patterns** | **0.74** | **0.41** | **0.43** | **0.79** | **0.83** | **0.82** | **0.76** | **0.75** | **0.79** |

Table 4: Comparison of our model against various emotion recognition systems: **LIWC** uses a bag of words approach; **CNN$_{\text{w2v}}$** is the proposed CNN model and word vectors obtained from (Deriu et al., 2017); **char** refers to character-level features; **n-gram** employ unigrams, bigrams, and trigrams as features; **CNN$_{\text{BASIC}}$** uses the proposed CNN architecture with basic patterns; **EmoNet** (Abdul-Mageed and Ungar, 2017) and **DeepMoji** (Felbo et al., 2017) are state-of-the-art emotion recognition models.

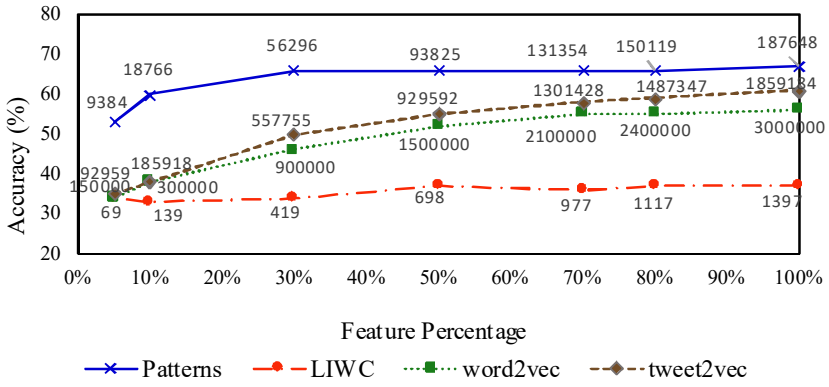


Figure 3: Consistency check for various affect resources.

| Model | Input | Epochs | Accuracy |
|---|---|---|---|
| **RNN$_{\text{w2v}}$** | word2vec (Mikolov et al., 2013) | 24 | 0.53 |
| **CNN$_{\text{char}}$** | character embeddings (end-to-end) | 50 | 0.63 |
| **CNN$_{\text{w2v}}$** | word vectors (Deriu et al., 2017) | 33 | **0.69** |
| **EmoNet** | word embeddings (end-to-end) | 23 | 0.58 |
| **DeepMoji** | word embeddings (end-to-end) | 100 | 0.63 |
| **BiGRNN** | our enriched patterns‡ | 12 | **0.68** |
| **CARER$_{\beta}$** | our enriched patterns‡ | 12 | **0.72** |
| **CARER$_{\text{EK}}$** | our enriched patterns | 12 | **0.81** |

Table 5: Comparison of our method against deep learning models, using Ekman's 6 emotions and the accuracy metric.

due to the way they were constructed. High coverage also means that the enriched patterns should demonstrate stability, in terms of how useful they are in an emotion classification task, even when reduced to smaller sizes. There are two cases where this could be useful: *limited data* and *limited computing resources*. Therefore, to test for pattern consistency, we randomly selected several pattern sizes[15] and trained a random forest classifier using the eight emotions dataset. This model performs comparable to **CARER$_{\beta}$** (average F1-score of 65%), and it has the benefit of faster training time, making it suitable for the aforementioned experiment. We compared with the results obtained from the **LIWC** lexicon (affect dimension) (Pennebaker et al., 2007), **word2vec** (Mikolov et al., 2013), and **tweet2vec** (Deriu et al., 2017).[16]

As shown in Figure 3, due to the limited coverage of the **LIWC** lexicon, such resources may not be feasible on evolving, large-scale datasets. In contrast, **word2vec** contains over 3 million unique word embeddings and has been proven effective for text classification. However, if we keep reducing the available word vectors of **word2vec**, which is common when there are limited computing resources, the accuracy keeps dropping at significant rates. **tweet2vec** has a similar effect. In the case of our patterns, the classification results remain relatively stable, even when reducing the patterns to 30% and 10% of the original size. These results show that the proposed features are feasible to address the text-based emotion recognition problem. Moreover, the patterns are highly beneficial where

[15] We employed the primitive patterns‡ used in CARER$_{\beta}$.

[16] Models were trained using the random forest implementation (depth=15 and estimators=50) provided by scikit-learn.

| | Document | GT | DeepMoji | EmoNet | CARER$_{EK}$ | Enriched Patterns |
|---|---|---|---|---|---|---|
| **Short text** | damn what a **night** | joy | surprise | sadness | joy | **what a** { *day*, *rush*, *pass* } |
| | **want** it to snow | joy | sadness | fear | joy | { *need*, *hoping* } **it** |
| **Rare words** | **whaaaaaat.** i did not expect that at all | surprise | sadness | fear | surprise | { *wondering*, *what* } **i** |
| **Mixed emotions** | got **thee** worst sleep ever | anger | sadness | sadness | anger | **got** { *madd*, *thatt*, *bacc* } |
| | what the fuck is **going** on !? | fear | anger | sadness | fear | **is** { *ends*, *finishes* } |

Table 6: Classified documents in the testing data. Words in **bold blue** correspond to the $sw$ component of one of the enriched patterns extracted from the document. Words in *italic green* represent other relevant word examples found in the cluster $sw$ belongs to. Words in **bold pink** denote the connector word/s $cw$ in the pattern; E.g., *"it"* is $cw$ of patterns *"need it"* and *"hoping it"*. **GT** stands for ground truth. **DeepMoji**, **EmoNet**, and **CARER$_{EK}$** correspond to the models reported in Table 5.

there is shortage of computing and linguistic resources.

### 6.2 What's captured by CARER?

In Table 6, we provide samples extracted from the testing data. The examples show different cases where the comparison models struggled to capture important contextual information that helps to determine the emotion conveyed in the text. For instance, in the short text, *"damn what a night"*, only our model was able to interpret the statement as *joy* because it uses the *"what a"* pattern and its corresponding subject words to determine that this statement has a stronger association with *joy*. Our model also works well for capturing *rare words* and for *disambiguating* emotional meaning using the enriched and refined contextual information of the patterns. Rare words like *"whaaaaaat"* and *"thee"* help to implicitly convey intense emotional expressions, which are also captured and considered important by our enriched patterns. Emotion-relevant verbs, such as *"want"* and *"going"* are also considered important context that help to convey and interpret emotion. Overall, the enriched patterns efficiently capture important emotional information that other models seem to ignore.

## 7 Conclusion

We proposed a graph-based feature extraction mechanism to extract emotion-relevant representations in an unsupervised manner. The contextualized affect representations are further enriched with word embeddings and are used to train several deep learning-based emotion recognition models. The patterns capture implicit and explicit linguistic emotional information which significantly improves emotion recognition results.

We offered a detailed analysis demonstrating special cases where the patterns are helpful to further extract and understand emotional information from textual information. For instance, short text is a challenging problem in emotion recognition and various natural language tasks; the proposed contextualized patterns show promising results in addressing this issue by helping the models to capture nuanced information which is useful to determine the overall emotion expressed in a piece of text. The proposed method paves the way for building more interpretable emotion recognition systems which have various implications when investigating human behavioural data (Saravia et al., 2015, 2016b; Chang et al., 2016) and building empathy-aware conversational agents.

In the future work, we aim to investigate the graph-based patterns more in-depth and provide a more comprehensive and advanced theoretical discussion of how they are constructed. We also hope to keep improving the pattern weighting mechanism so as to improve the overall performance on emotion recognition tasks and minimize trade-off between pattern coverage and performance. We plan to employ transfer learning methods with the proposed enriched patterns and test on other emotion-related problems such as sentiment classification and sarcasm detection. The proposed methodology is also being expanded to support Spanish and Japanese emotion recognition tasks.

## Acknowledgements

This research was supported by the Ministry of Science and Technology (#106-2221-E-007-115-MY2 and #106-3114-E-007-013).

## References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proc. of ACL*.

Karin Becker, Viviane Moreira, and Aline 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, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proc. of ICML*.

Chun-Hao Chang, Elvis Saravia, and Yi-Shin Chen. 2016. Subconscious crowdsourcing: A feasible data collection mechanism for mental disorder detection on social media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 374–379. IEEE.

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

Cindy Chung and James Pennebaker. 2007. The psychological functions of function words. *Social communication*, 1:343–359.

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 *Proc. of ICWSM*.

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. In *Proc of EMNLP*.

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

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *Proc. of ACL*.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*.

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

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of massive datasets*. Cambridge university press.

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

Saif M Mohammad. 2012. Emotional tweets. In *Proc. of ACL*.

Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Vinod Nair and Geoffrey Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proc. of ICML*.

Febian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

James Pennebaker, Roger Booth, and Martha 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 *Proc. of ICDM*.

Ashequl Qadir and Ellen Riloff. 2013. Bootstrapped learning of emotion hashtags# hashtags4you. In *Proc. of Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proc. of ACL*.

Kirk Roberts, Michael Roach, Joseph Johnson, Josh Guthrie, and Sanda Harabagiu. 2012. Empatweet: Annotating and detecting emotions on twitter. In *Proc. of LREC*.

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

Elvis Saravia, Carlos Argueta, and Yi-Shin Chen. 2015. Emoviz: Mining the world's interest through emotion analysis. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 753–756. IEEE.

Elvis Saravia, Carlos Argueta, and Yi-Shin Chen. 2016a. Unsupervised graph-based pattern extraction for multilingual emotion classification. *Social Network Analysis and Mining*, 6(1):92.

Elvis Saravia, Chun-Hao Chang, Renaud Jollet De Lorenzo, and Yi-Shin Chen. 2016b. Midas: Mental illness detection and analysis via social media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 1418–1421. IEEE.

Julio Savigny and Ayu Purwarianti. 2017. Emotion classification on youtube comments using word embedding. In *Proc. of ICAICTA*.

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*.

Carlo Strapparava, Alessandro Valitutti, et al. 2004. Wordnet affect: an affective extension of wordnet. In *Proc. of LREC*.

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*.

Svitlana Volkova and Yoram Bachrach. 2016. Inferring perceived demographics from user emotional tone and user-environment emotional contrast. In *Proc. of ACL*.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In *Proc. of ACL*.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing twitter" big data" for automatic emotion identification. In *Proc. of SocialCom*.

Joe Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. of NIPS*.