applied sciences

MDPI

Article

A Deep Learning-Based Approach for Multi-Label Emotion Classification in Tweets

Mohammed Jabreel [1,2,*] and Antonio Moreno [1]

[1] ITAKA Research Group, Universitat Rovira i Virgili, 43007 Tarragona, Spain; antonio.moreno@urv.cat
[2] Department of Computer Science, Hodeidah University, 1821 Hodeidah, Yemen
* Correspondence: mhjabreel@gmail.com

Received: 15 February 2019; Accepted: 12 March 2019; Published: 17 March 2019

Abstract: Currently, people use online social media such as Twitter or Facebook to share their emotions and thoughts. Detecting and analyzing the emotions expressed in social media content benefits many applications in commerce, public health, social welfare, etc. Most previous work on sentiment and emotion analysis has only focused on single-label classification and ignored the co-existence of multiple emotion labels in one instance. This paper describes the development of a novel deep learning-based system that addresses the multiple emotion classification problem in Twitter. We propose a novel method to transform it to a binary classification problem and exploit a deep learning approach to solve the transformed problem. Our system outperforms the state-of-the-art systems, achieving an accuracy score of 0.59 on the challenging SemEval2018 Task 1:E-cmulti-label emotion classification problem.

Keywords: opinion mining; sentiment analysis; emotion classification; deep learning; Twitter

1. Introduction

Emotions are the key to people's feelings and thoughts. Online social media, such as Twitter and Facebook, have changed the language of communication. Currently, people can communicate facts, opinions, emotions, and emotion intensities on different kinds of topics in short texts. Analyzing the emotions expressed in social media content has attracted researchers in the natural language processing research field. It has a wide range of applications in commerce, public health, social welfare, etc. For instance, it can be used in public health [1,2], public opinion detection about political tendencies [3,4], brand management [5], and stock market monitoring [6]. Emotion analysis is the task of determining the attitude towards a target or topic. The attitude can be the polarity (positive or negative) or an emotional state such as joy, anger, or sadness [7].

Recently, the multi-label classification problem has attracted considerable interest due to its applicability to a wide range of domains, including text classification, scene and video classification, and bioinformatics [8]. Unlike the traditional single-label classification problem (i.e., multi-class or binary), where an instance is associated with only one label from a finite set of labels, in the multi-label classification problem, an instance is associated with a subset of labels.

Most previous work on sentiment and emotion analysis has only focused on single-label classification. Hence, in this article, we focus on the multi-label emotion classification task, which aims to develop an automatic system to determine the existence in a text of none, one, or more out of eleven emotions: the eight Plutchik [9] categories (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation) that are shown in Figure 1, plus optimism, pessimism, and love.

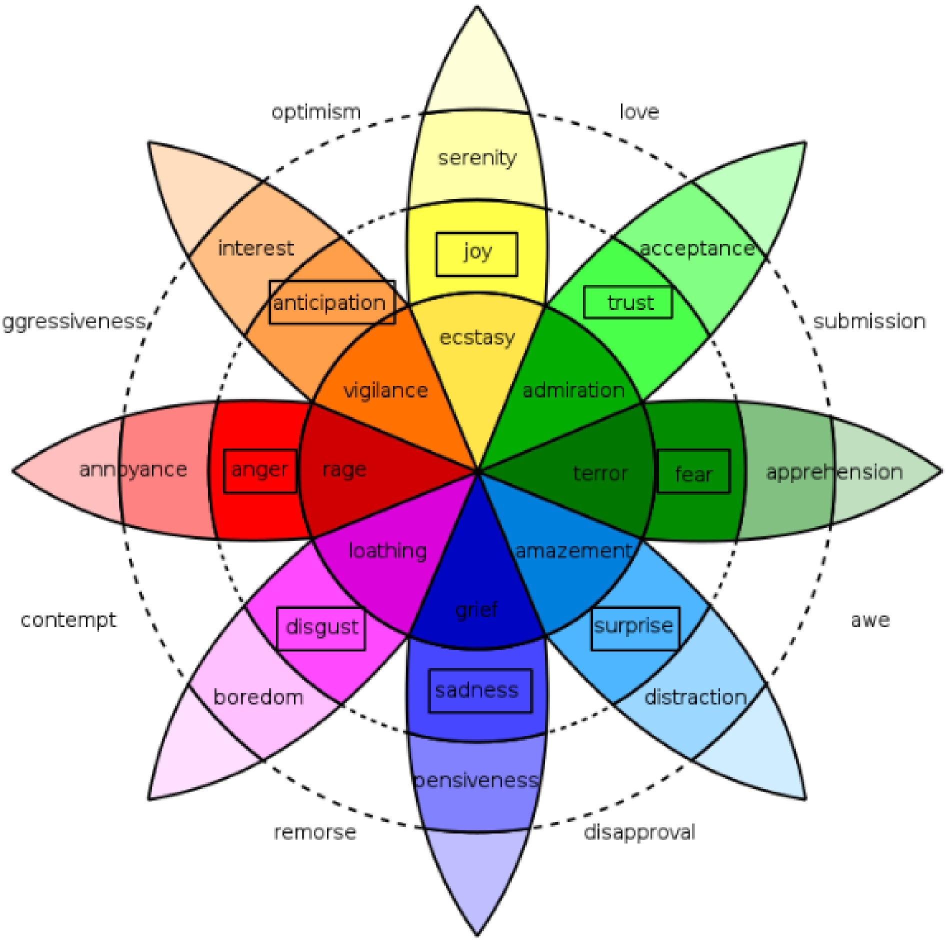


Figure 1. The set of the eight basic emotions proposed by Plutchik [9].

One of the most common approaches to addressing the problem of multi-label classification is the *problem transformation*. With this approach, a multi-label problem is transformed into one or more single-label (i.e., binary or multi-class) problems. Specifically, single-label classifiers are learned and employed; after that, the classifiers' predictions are transformed into multi-label predictions.

Different transformation methods have been proposed in the multi-label literature. The most common method is called *binary relevance* [10,11]. The idea of the binary relevance method is simple and intuitive. A multi-label problem is transformed into multiple binary problems, one problem for each label. Then, an independent binary classifier is trained to predict the relevance of one of the labels. Although binary relevance is popular in the literature, due to its simplicity, it suffers from directly modeling correlations that may exist between labels. However, it is highly resistant to overfitting label combinations, since it does not expect examples to be associated with previously-observed combinations of labels.

In this article, we propose a novel transformation method, called *xy*-pair-set, for the multi-label classification problem. Unlike binary relevance methods, our method transforms the problem into only one binary classification problem as described in Section 3. Additionally, we exploit the successes of deep learning models, especially the word2vecmethods' family [12] and the recurrent neural networks [13,14] and attention models [15,16], to develop a system that solves the transformed binary classification problem. The critical component of our system is the embedding module, which uses three embedding models and an attention function to model the relationship between the input and the label.

To summarize, the contribution of this work is four-fold.

- We propose a novel transformation mechanism for the multi-label classification problem.
- We propose a novel, attentive deep learning system, which we call Binary Neural Network (BNet), which works on the new transformation method. Our system is a data-driven, end-to-end neural-based model, and it does not rely on external resources such as parts of speech taggers and sentiment or emotion lexicons.
- We evaluate the proposed system on the challenging multi-label emotion classification dataset of SemEval-2018 Task1: Affect in Tweets.
- The experimental results show that our system outperforms the state-of-the-art systems.

The rest of the article is structured as follows. In Section 2, we overview the related work on multi-label problem transformation methods and Twitter sentiment and emotion analysis. In Section 3, we explain in detail the methodology. In Section 4, we report the experimental results. In Section 5, the conclusions and the future work are presented.

2. Related Works

In this section, we overview the most popular research studies related to this work. In Section 2.1, we summarize the most common multi-label problem transformation methods. Section 2.2 gives an overview of the state-of-the-art works on the problem of multi-label emotion classification on Twitter.

2.1. Problem Transformation Methods

Let $X = \{x_1, x_2, \ldots x_n\}$ be the set of all instances and $Y = \{y_1, y_2, \ldots, y_m\}$ be the set of all labels. We can define the set of data:

$$D = \{(x_i, \hat{Y}_i) | x_i \in X \text{ and } \hat{Y}_i \subseteq Y \text{ is the set of labels associated with } x_i\} \quad (1)$$

In this expression, D is called a supervised multi-label dataset.

The task of multi-label classification is challenging because the number of label sets grows exponentially as the number of class labels increases. One common strategy to address this issue is to transform the problem into a traditional classification problem. The idea is to simplify the learning process by exploiting label correlations. Based on the order of the correlations, we can group the existing transformation methods into three approaches [17,18], namely first-order approaches, second-order approaches, and high-order approaches.

First-order approaches decompose the problem into some independent binary classification problems. In this case, one binary classifier is learned for each possible class, ignoring the co-existence of other labels. Thus, the number of independent binary classifiers needed is equal to the number of labels. For each multi-label training example $(x_i, \hat{Y}_i) \in D$, $y_k \in Y$, we construct a binary classification training set, D_k as the following: x_i will be regarded as one positive example if $y_k \in \hat{Y}_i$ and one negative example otherwise. In the first case, we will get a training example in the form $(x_i, 1) \in D_k$, which will be $(x_i, 0) \in D_k$ in the second case. Thus, for all labels $\{y_1, y_2, \ldots, y_m\} \in Y$, m training sets $\{D_1, D_2, \ldots, D_m\}$ are constructed. Based on that, for each training set D_k, one binary classifier can be learned with popular learning techniques such as AdaBoost [19], k-nearest neighbor [20], decision trees, random forests [21,22], etc. The main advantage of first-order approaches is their conceptual simplicity and high efficiency. However, these approaches can be less effective due to their ignorance of label correlations.

Second-order approaches try to address the lack of modeling label correlations by exploiting pairwise relationships between the labels. One way to consider pairwise relationships is to train one binary classifier for each *pair* of labels [23]. Although second-order approaches perform well in several domains, they are more complicated than the first-order approaches in terms of the number of classifiers. Their complexity is quadratic, as the number of classifiers needed is $\binom{m}{2}$. Moreover, in real-world applications, label correlations could be more complex and go beyond second-order.

High-order approaches tackle the multi-label learning problem by exploring high-order relationships among the labels. This can be fulfilled by assuming linear combinations [24], a nonlinear mapping [25,26], or a shared subspace over the whole label space [27]. Although high-order approaches have stronger correlation-modeling capabilities than their first-order and second-order counterparts, these approaches are computationally demanding and less scalable.

Our transformation mechanism, shown in Section 3.1, is a simple as the first-order approaches and can model, implicitly, high-order relationships among the labels if some requirements, detailed in Section 3.1, are fulfilled. It requires only one binary classifier, and the number of training examples grows polynomially in terms of the number of instances and the number of labels. If the number of training examples in the multi-label training dataset is n and the number of the labels is m, then the number of the training examples in the transformed binary training set is $n \times m$.

2.2. Emotion Classification in Tweets

Various machine learning approaches have been proposed for traditional emotion classification and multi-label emotion classification. Most of the existing systems solve the problem as a text classification problem. Supervised classifiers are trained on a set of annotated corpora using a different set of hand-engineered features. The success of such models is based on two main factors: a large amount of labeled data and the intelligent design of a set of features that can distinguish between the samples. With this approach, most studies have focused on engineering a set of efficient features to obtain a good classification performance [28–30]. The idea is to find a set of informative features to reflect the sentiments or the emotions expressed in the text. Bag-of-Words (BoW) and its variation, n-grams, is the representation method used in most text classification problems and emotion analysis. Different studies have combined the BoW features with other features such as the parts of speech tags, the sentiment and the emotion information extracted from lexicons, statistical information, and word shapes to enrich the text representation.

Although BoW is a popular method in most text classification systems, it has some drawbacks. Firstly, it ignores the word order. That means that two documents may have the same or a very close representation as far as they have the same words, even though they carry a different meaning. The n-gram method resolves this disadvantage of BoW by considering the word order in a context of length n. However, it suffers from sparsity and high dimensionality. Secondly, BoW is scarcely able to model the semantics of words. For example, the words *beautiful*, *wonderful* and *view* have an equal distance in BoW, where the word *beautiful* is closer to the word *wonderful* than the word *view* in the semantic space.

Sentiment and emotion lexicons play an essential role in developing efficient sentiment and emotion analysis systems. However, it is difficult to create such lexicons. Moreover, finding the best combination of lexicons in addition to the best set of statistical features is a time-consuming task.

Recently, deep learning models have been utilized to develop end-to-end systems in many tasks including speech recognition, text classification, and image classification. It has been shown that such systems automatically extract high-level features from raw data [31,32].

Baziotis et al. [33], the winner of the multi-label emotion classification task of SemEval-2018 Task1: Affect in Tweets, developed a bidirectional Long Short-Term Memory (LSTM) with a deep attention mechanism. They trained a word2vec model with 800,000 words derived from a dataset of 550 million tweets. The second place winner of the SemEval leaderboard trained a word-level bidirectional LSTM with attention, and it also included non-deep learning features in its ensemble [34]. Ji Ho Park et al. [35] trained two models to solve this problem: regularized linear regression and logistic regression classifier chain [11]. They tried to exploit labels' correlation to perform multi-label classification. With the first model, the authors formulated the multi-label classification problem as a linear regression with label distance as the regularization term. In their work, the logistic regression classifier chain method was used to capture the correlation of emotion labels. The idea is to treat the multi-label problem as a sequence of binary classification problems by taking the prediction of the previous classifier as an extra input to the next classifier.

In this work, we exploited the deep learning-based approach to develop a system that can extract a high-level representation of the tweets and model an implicit high-order relationship among the labels. We used the proposed system alongside the proposed transformation method to train a function that can solve the problem of multi-label emotion classification in tweets. The next section explains the details of our proposed system.

3. Methodology

This section shows the methodology of this work. First, we explain in Section 3.1 the proposed transformation method, *xy*-pair-set. Afterwards, we describe the proposed system in Section 3.2.

3.1. xy-Pair-Set: Problem Transformation

The proposed transformation method *xy*-pair-set transforms a multi-label classification dataset D into a supervised binary dataset $\hat{D}$ as follows:

$$\begin{aligned} &\forall x_i \in X, y \in Y \text{ and } (x_i, \hat{Y}_i) \in D, \exists!((x_i, y), \phi) \in \hat{D} \\ &\text{where } \phi = \begin{cases} 1 & \text{if } y \in \hat{Y}_i \\ 0 & otherwise \end{cases} \end{aligned} \tag{2}$$

Algorithm 1 explains the implementation of the proposed transformation method. It takes as inputs a multi-label dataset D (Equation (1)) and a set of labels Y, and it returns a transformed binary dataset. We show next an illustrative example.

Algorithm 1: *xy*-pair-set algorithm.

```
Input: Input: a multi-label classification dataset $D$ and a set of labels $Y$
Output: Output: a binary classification dataset $\hat{D}$
$\hat{D} = \{\}$;
foreach $(x_i, \hat{Y}_i) \in D$ do
    foreach $y \in Y$ do
        $\hat{x}_i = (x_i, y)$ ;                    ▷ a tuple of $x_i$ and $y$.
        if $y \in \hat{Y}_i$ then
            $\hat{D} = \hat{D} \bigcup (\hat{x}_i, 1)$ ;
        else
            $\hat{D} = \hat{D} \bigcup (\hat{x}_i, 0)$ ;
        end
    end
end
return $\hat{D}$
```

Let $X = \{x1, x2\}$, $Y = \{a, b, c\}$ and $D = \{(x1, \{a, c\}), (x2, \{b\})\}$. The output of the binary relevance transformation method is a set of three independent binary datasets, one for each label. That is, $D_a = \{(x1, 1), (x2, 0)\}$, $D_b = \{(x1, 0), (x2, 1)\}$, and $D_c = \{(x1, 1), (x2, 0)\}$. In contrast, the output of our transformation method is a single binary dataset $\hat{D} = \{((x1, a), 1), ((x1, b), 0), ((x1, c), 1), ((x2, a), 0), ((x2, b), 1), ((x2, c), 0)\}$.

The task in this case, unlike the traditional supervised binary classification algorithms, is to develop a learning algorithm to learn a function $g : X \times Y \to \{0, 1\}$. The success of such an algorithm is based on three requirements: (1) an encoding method to represent an instance $x \in X$ as a high-dimensional vector V_x, (2) a method to encode a label $y \in Y$ as a vector V_y, and (3) a method to represent the relation between the instance x and the label y. These three conditions make g able to capture the relationships inputs-to-labels and labels-to-labels. In this work, we take advantage of the successes of deep learning models to fulfill the three requirements listed above. We empirically show the success of our system with respect to these conditions as reported in Sections 4.6 and 4.7.

3.2. BNet: System Description

This subsection explains the proposed system to solve the transformed binary problem mentioned above. Figure 2 shows the graphical depiction of the system's architecture. It is composed of three parts: the embedding module, the encoding module, and the classification module. We explain in detail each of them below.

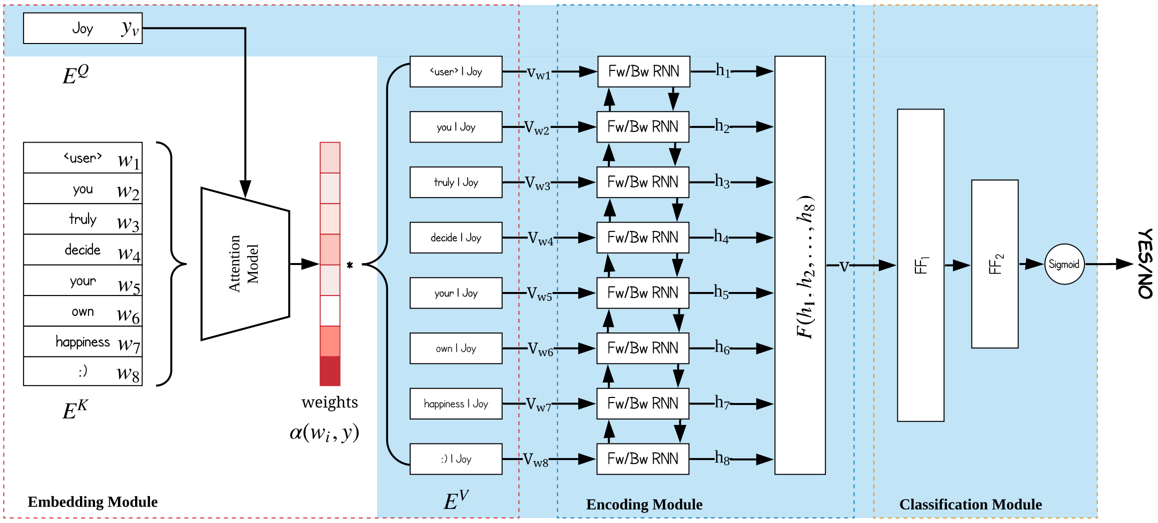


Figure 2. An illustration of the proposed system. Shaded parts are trainable. Fw and Bw refer to the Forward and Backward cells respectively, and FF means Feed-Forward layer.

3.2.1. Embedding Module

Let (W, y) be the pair of inputs to our system, where $W = \{w_1, w_2, \ldots, w_l\}$ is the set of the words in a tweet and y is the label corresponding to an emotion. The goal of the embedding module is to represent each word w_i by a vector v_{w_i} and the label by a vector v_y.

Our embedding module can be seen as a function that maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The query is the trainable label embedding, $E^Q(y)$; the keys are the pretrained words embeddings, $E^K(w_i)$ $\forall w_i \in W$; and the values are the trainable words embeddings, $E^V(w_i)$ $\forall w_i \in W$.

As shown in Figure 2, we used the output of E^Q and E^K as inputs to the attention model to find the alignments, i.e., the weights α, between the label y and the words W of the input tweet. This step models the relation between the input and the label. As soon as the weights were obtained, we then multiplied each word's vector that came from the embedding E^V by its corresponding weight. Given that, the final representation of a word $w_i \in W$ is given as the following:

$$v_{w_i} = E^V(w_i) \cdot \alpha(w_i, y) \tag{3}$$

The function α is an attention-based model, which finds the strength of the relationship between the word w_i and the label y based on their semantic similarity. That is, $\alpha(w_i, y)$ is a value based on the distance $\Delta(w_i, y)$ between w_i and y as:

$$\alpha(w_i, y) = \frac{e^{\Delta(w_i,y)}}{\sum_{w_j \in W} e^{\Delta(w_j,y)}} \tag{4}$$

Here, $\Delta(w_i, y)$ is a scalar score that represents the similarity between the word w_i and the label y:

$$\Delta(w_i, y) = E^k(w_i) \cdot {v_y}^T \tag{5}$$

$$v_y = E^Q(y) \tag{6}$$

It is worth noting that $\alpha(w_i, y) \in [0, 1]$ and:

$$\sum_{w_i \in W} \alpha(w_i, y) = 1 \tag{7}$$

3.2.2. Encoding Module

The goal of the encoding module is to map the sequence of word representations $\{v_{w_1}, v_{w_2}, \ldots, v_{w_l}\}$ that is obtained from the embedding module to a single real-valued dense vector. In this work, we used a Recurrent Neural Network (RNN) to design our encoder. RNN reads the input sequence of vectors in a forward direction (left-to-right) starting from the first symbol v_{w_1} to the last one v_{w_l}. Thus, it processes sequences in temporal order, ignoring the future context. However, for many tasks on sequences, it is beneficial to have access to future, as well as to past information. For example, in text processing, decisions are usually made after the whole sentence is known. The Bidirectional Recurrent Neural Network (BiRNN) variant [13] proposed a solution for making predictions based on both past and future information.

A BiRNN consists of forward $\overrightarrow{\phi}$ and backward $\overleftarrow{\phi}$ RNNs. The first one reads the input sequence in a forward direction and produces a sequence of forward hidden states $(\overrightarrow{h_1}, \ldots, \overrightarrow{h_l})$, whereas the former reads the sequence in the reverse order $(v_{w_l}, \ldots, v_{w_1})$, resulting in a sequence of backward hidden states $(\overleftarrow{h_l}, \ldots, \overleftarrow{h_1})$.

We obtained a representation for each word v_{w_t} by concatenating the corresponding forward hidden state $\overrightarrow{h_t}$ and the backward one $\overleftarrow{h_t}$. The following equations illustrate the main ideas:

$$\overrightarrow{h_t} = \overrightarrow{\phi}(v_{w_t}, \overrightarrow{h_{t-1}}) \tag{8}$$

$$\overleftarrow{h_t} = \overleftarrow{\phi}(v_{w_t}, \overleftarrow{h_{t+1}}) \tag{9}$$

$$h_t = [\, \overrightarrow{h_t}; \overleftarrow{h_t} \,] \tag{10}$$

The final input representation of the sequence is:

$$c = F(\{h_1, h_2, \ldots, h_l\}) \tag{11}$$

We simply chose F to be the last hidden state (i.e., $F(\{h_1, h_2, \ldots, h_n\}) = h_n$).

In this work, we used two Gated Recurrent Units (GRUs) [14], one as $\overrightarrow{\phi}$ and the other as $\overleftarrow{\phi}$. This kind of RNN was designed to have more persistent memory, making them very useful to capture long-term dependencies between the elements of a sequence. Figure 3 shows a graphical depiction of a gated recurrent unit.

A GRU has *reset* (r_t) and *update* (z_t) gates. The former can completely reduce the past hidden state h_{t-1} if it finds that it is irrelevant to the computation of the new state, whereas the later is responsible for determining how much of h_{t-1} should be carried forward to the next state h_t.

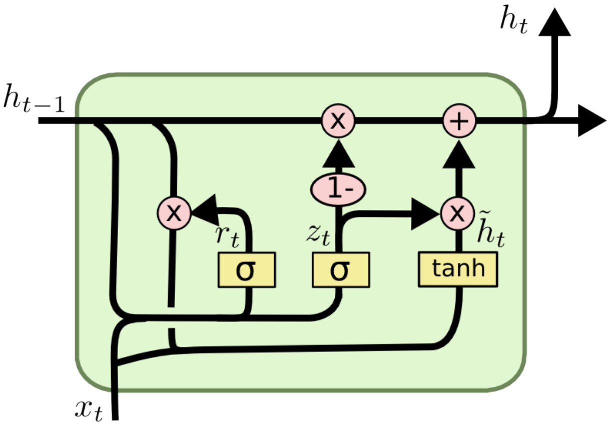


Figure 3. Gated Recurrent Unit (GRU).

The output h_t of a GRU depends on the input x_t and the previous state h_{t-1}, and it is computed as follows:

$$r_t = \sigma\left(W_r \cdot [h_{t-1}; x_t] + b_r\right) \tag{12}$$

$$z_t = \sigma\left(W_z \cdot [h_{t-1}; x_t] + b_z\right) \tag{13}$$

$$\widetilde{h}_t = tanh\left(W_h \cdot [(r_t \odot h_{t-1}); x_t] + b_h\right) \tag{14}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \tag{15}$$

In these expressions, r_t and z_t denote the *reset* and *update* gates, $\widetilde{h}_t$ is the candidate output state, and h_t is the actual output state at time t. The symbol $\odot$ stands for element-wise multiplication; σ is a sigmoid function; and; stands for the vector-concatenation operation. $W_r, W_z, W_h \in \mathbb{R}^{d_h \times (d+d_h)}$ and $b_r, b_z, b_h \in \mathbb{R}^{d_h}$ are the parameters of the *reset* and *update* gates, where d_h is the dimension of the hidden state and d is the dimension of the input vector.

3.2.3. Classification Module

Our classifier was composed of two feed-forward layers with the *ReLU* activation function followed by a Sigmoid unit.

4. Experiments and Results

In this section, we first describe the experimental details, and then, we describe the dataset and the pre-processing we used. Afterwards, we introduce the state-of-the-art systems we compared our system with, and finally, we report the empirical validation proving the effectiveness of our system.

4.1. Experimental Details

Table 1 shows the hyperparameters of our system, which was trained using Adam [36], with a learning rate of 0.0001, β_1 = 0.5, and a mini-batch size of 32 to minimize the binary cross-entropy loss function:

$$\mathcal{L}(\theta, \hat{D}) = -\mathbb{E}_{((x_i,y_i),\phi_i)\sim\hat{D}}\left[\phi_i \cdot g(x_i, y_i) + (1 - \phi_i) \cdot (1 - g(x_i, y_i))\right] \tag{16}$$

where, $g(x_i, y_i)$ is the predicted value, ϕ_i is the real value, and θ is the model's parameters.

The hyperparameters of our system were obtained by applying Bayesian optimization [37]. We used the development set as a validation set to fine-tune those parameters.

Table 1. Hyperparameters of our system.

Parameter	Value
Embedding Module	E^Q: Dimensions: 11 × 310 Initialization: Uniform (−0.02, 0.02) Trainable: Yes
	E^K: Dimensions: 13,249 × 310 Initialization: Pretrained model (we used the pretrained embeddings provided in [33]). Trainable: No
	E^V: Dimensions: 13,249 × 310 Initialization: Uniform (−0.02, 0.02) Trainable: Yes
Encoding Module	RNN Cell: GRU Hidden size: 200 Layers: 2 Encoding: last hidden state RNN dropout: 0.3
Classification Module	FF1: 1024 units FF2: 512 units Sigmoid: 1 unit Activation: ReLU Dropout: 0.3

4.2. Dataset

In our experiments, we used the multi-label emotion classification dataset of SemEval-2018 Task1: Affect in Tweets [30]. It contains 10,983 samples divided into three splits: training set (6838 samples), validation set (886 samples), and testing set (3259 samples). For more details about the dataset, we refer the reader to [38]. We trained our system on the training set and used the validation set to fine-tune the parameters of the proposed system. We pre-processed each tweet in the dataset as follows:

- Tokenization: We used an extensive list of regular expressions to recognize the following meta information included in tweets: Twitter markup, emoticons, emojis, dates, times, currencies, acronyms, hashtags, user mentions, URLs, and words with emphasis.
- As soon as the tokenization was done, we lowercased words and normalized the recognized tokens. For example, URLs were replaced by the token "<URL>", and user mentions were replaced by the token "<USER>". This step helped to reduce the size of the vocabulary without losing information.

4.3. Comparison with Other Systems

We compared the proposed system with the state-of-the-art systems used in the task of multi-label emotion classification, including:

- SVM-unigrams: a baseline support vector machine system trained using just word unigrams as features [30].
- NTUA-SLP: the system submitted by the winner team of the SemEval-2018 Task1:E-cchallenge [33].
- TCS: the system submitted by the second place winner [34].
- PlusEmo2Vec: the system submitted by the third place winner [35].
- Transformer: a deep learning system based on large pre-trained language models developed by the NVIDIA AI lab [39].

4.4. Evaluation Metrics

We used multi-label accuracy (or Jaccard index), the official competition metric used by the organizers of SemEval-2018 Task 1: Affect in Tweets, for the E-c sub task, which can be defined as the size of the intersection of the predicted and gold label sets divided by the size of their union.

$$Jaccard = \frac{1}{|T|} \sum_{t \in T} \frac{G_t \cap P_t}{G_t \cup P_t} \tag{17}$$

In this expression, G_t is the set of the gold labels for tweet t, P_t is the set of the predicted labels for tweet t, and T is the set of tweets. Additionally, we also used the micro-averaged F-score and the macro-averaged F-score.

Let $\#_c(l)$ denote the number of samples correctly assigned to the label l, $\#_p(l)$ the number of samples assigned to l, and $\#(l)$ the number of actual samples in l. The micro-averaged F1-score is calculated as follows:

$$P_{micro} = \frac{\sum_{l \in L} \#_c(l)}{\sum_{l \in L} \#_p(l)} \tag{18}$$

$$R_{micro} = \frac{\sum_{l \in L} \#_c(l)}{\sum_{l \in L} \#(l)} \tag{19}$$

$$F1_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}} \tag{20}$$

Thus, P_{micro} is the micro-averaged precision score, and R_{micro} is the micro-averaged recall score.

Let P_l, R_l, and F_l denote the precision score, recall score, and the F1-score of the label l. The macro-averaged F1-score is calculated as follows:

$$P_l = \frac{\#_c(l)}{\#_p(l)} \tag{21}$$

$$R_l = \frac{\#_c(l)}{\#(l)} \tag{22}$$

$$F_l = \frac{2 \times P_l \times R_l}{P_l + R_l} \tag{23}$$

$$F1_{macro} = \frac{1}{|L|} \sum_{l \in L} F_l \tag{24}$$

4.5. Results

We submitted our system's predictions to the SemEval Task1:E-C challenge. The results were computed by the organizers on a golden test set, for which we did not have access to the golden labels.

Table 2 shows the results of our system and the results of the compared models (obtained from their associated papers). As can be observed from the reported results, our system achieved the top Jaccard index accuracy and macro-averaged F1 scores among all the state-of-the-art systems, with a competitive, but slightly lower score for the micro-average F1.

To get more insight about the performance of our system, we calculated the precision score, the recall score, and the F1 score of each label. The results of this analysis are shown in Figure 4. We found that our system gave the best performance on the "joy" label followed by the "anger", "fear", "disgust", and "optimism" labels. The obtained F1-score of these labels was above 70%. The worst performance was obtained on the "trust", "surprise", "anticipation", and "pessimism" labels. In most cases, our system gave a recall score higher than the precision score. It seems that the system was aggressive against the emotions "trust", "surprise", "anticipation", and "pessimism" (i.e., the system associated a low number of samples to these labels). This can be attributed to the low number of training examples for these emotions and to the Out-Of-Vocabulary (OOV) problem.

Table 2. Results of our system and state-of-the-art systems. The best values are in bold.

Model	Accuracy (Jaccard)	Micro F1	Macro F1
BNet(Our System)	**0.590**	0.692	**0.564**
SVM-Unigrams	0.442	0.57	0.443
Transformer	0.577	0.690	0.561
NTUA-SLP	0.588	**0.701**	0.528
TCS	0.582	0.693	0.530
PlusEmo2Vec	0.576	0.692	0.497

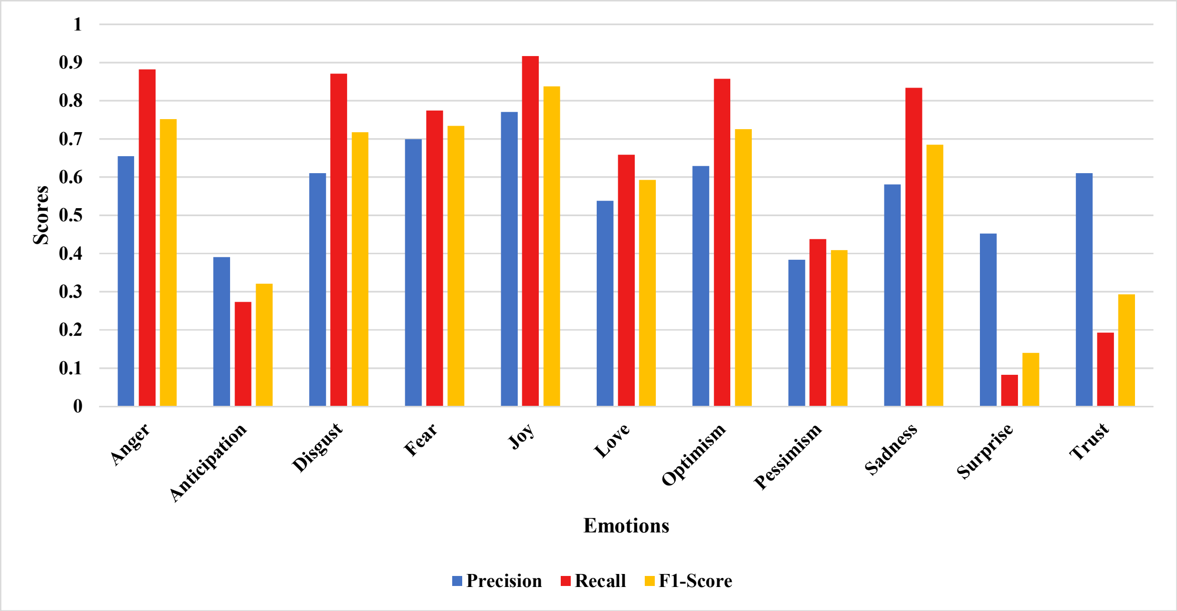
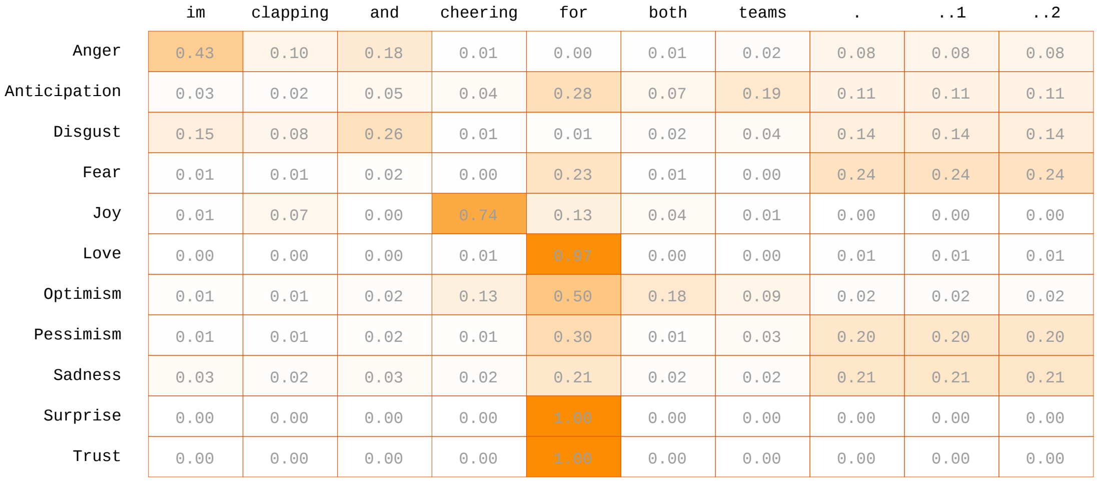


Figure 4. Performance analysis.

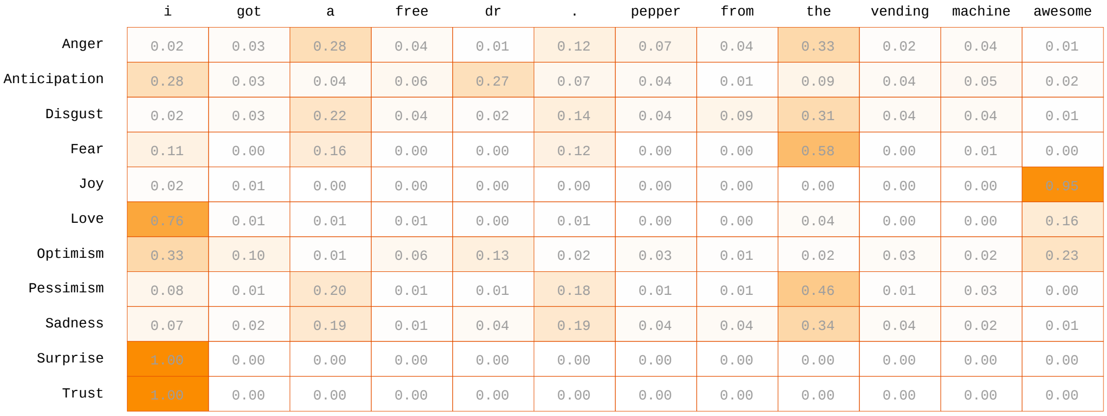
4.6. Attention Visualizations

We visualized the attention weights to get a better understanding of the performance of our system. The results are described in Figures 5–8, which show heat-maps of the attention weights on the top four example tweets from the validation set. The color intensity refers to the weight given to each word by the attention model. It represents the strength of the relationship between the word and the emotion, which reflects the importance of this word in the final prediction. We can see that the attention model gave the important weights to the common words, such as the stop words, in case the tweet was not assigned to the emotion; for example, the word "for" in Figure 5 and the word "this" in Figure 7 and the token "<user>" in Figure 8. Moreover, it also gives a high weight for the words and the emojis related to emotions (e.g., "cheering" and "awesome" for joy, "birthday" for love, etc.). An interesting observation is that when emojis were present, they were almost always selected as important if they were related to the emotion. For instance, we can see in Figure 7 that the sadness emotion relied heavily on the emoji. We also found that considering only one word to model the relation between the tweet and the emotions was not enough. In some cases, the emotion of a word may be flipped based on the context. For instance, consider the following tweet as an example: "When being #productive (doing the things that NEED to be done), #anxiety level decreases and #love level increases. #personalsexuality", the word "anxiety" is highly related to the emotion fear, but in this context, it shows optimism and trust emotions. However, our system misassociated this example with the fear emotion.



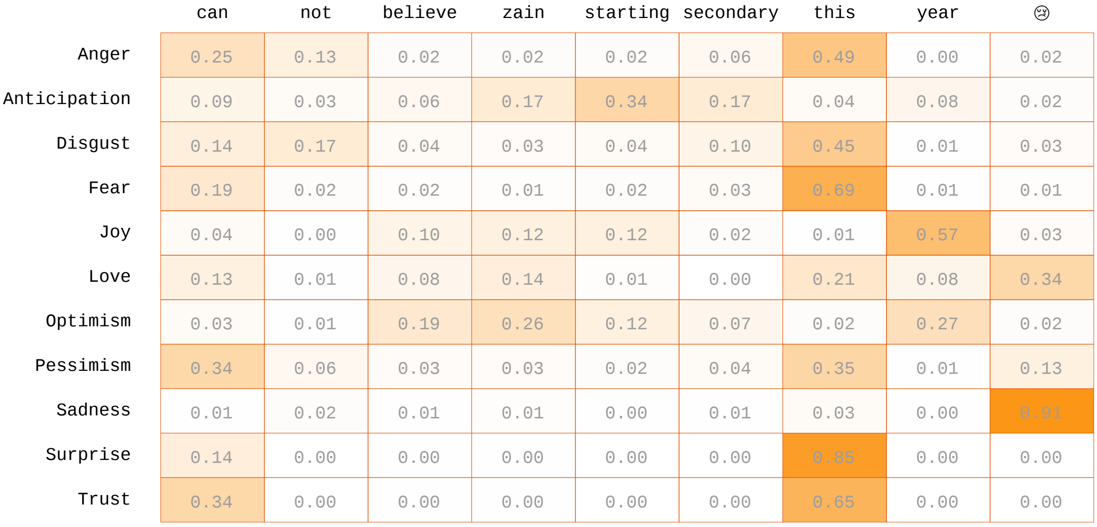
	im	clapping	and	cheering	for	both	teams	.	..1	..2
Anger	0.43	0.10	0.18	0.01	0.00	0.01	0.02	0.08	0.08	0.08
Anticipation	0.03	0.02	0.05	0.04	0.28	0.07	0.19	0.11	0.11	0.11
Disgust	0.15	0.08	0.26	0.01	0.01	0.02	0.04	0.14	0.14	0.14
Fear	0.01	0.01	0.02	0.00	0.23	0.01	0.00	0.24	0.24	0.24
Joy	0.01	0.07	0.00	0.74	0.13	0.04	0.01	0.00	0.00	0.00
Love	0.00	0.00	0.00	0.01	0.97	0.00	0.00	0.01	0.01	0.01
Optimism	0.01	0.01	0.02	0.13	0.50	0.18	0.09	0.02	0.02	0.02
Pessimism	0.01	0.01	0.02	0.01	0.30	0.01	0.03	0.20	0.20	0.20
Sadness	0.03	0.02	0.03	0.02	0.21	0.02	0.02	0.21	0.21	0.21
Surprise	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Trust	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00

Figure 5. Attention visualization example. Golden labels are {*joy*, *optimism*} and predicted labels are {*joy (0.91)*, *optimism (0.51)*}.



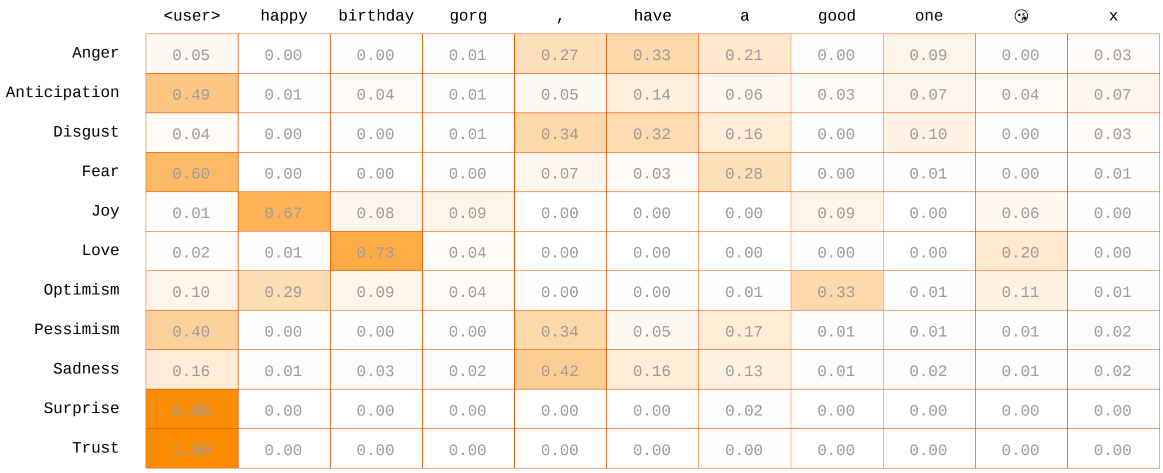
	i	got	a	free	dr	.	pepper	from	the	vending	machine	awesome
Anger	0.02	0.03	0.28	0.04	0.01	0.12	0.07	0.04	0.33	0.02	0.04	0.01
Anticipation	0.28	0.03	0.04	0.06	0.27	0.07	0.04	0.01	0.09	0.04	0.05	0.02
Disgust	0.02	0.03	0.22	0.04	0.02	0.14	0.04	0.09	0.31	0.04	0.04	0.01
Fear	0.11	0.00	0.16	0.00	0.00	0.12	0.00	0.00	0.58	0.00	0.01	0.00
Joy	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95
Love	0.76	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.04	0.00	0.00	0.16
Optimism	0.33	0.10	0.01	0.06	0.13	0.02	0.03	0.01	0.02	0.03	0.02	0.23
Pessimism	0.08	0.01	0.20	0.01	0.01	0.18	0.01	0.01	0.46	0.01	0.03	0.00
Sadness	0.07	0.02	0.19	0.01	0.04	0.19	0.04	0.04	0.34	0.04	0.02	0.01
Surprise	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Trust	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 6. Attention visualization example. Golden labels are {*joy*, *surprise*} and predicted labels are {*joy (0.97)*, *optimism (0.87)*}.



	can	not	believe	zain	starting	secondary	this	year	😢
Anger	0.25	0.13	0.02	0.02	0.02	0.06	0.49	0.00	0.02
Anticipation	0.09	0.03	0.06	0.17	0.34	0.17	0.04	0.08	0.02
Disgust	0.14	0.17	0.04	0.03	0.04	0.10	0.45	0.01	0.03
Fear	0.19	0.02	0.02	0.01	0.02	0.03	0.69	0.01	0.01
Joy	0.04	0.00	0.10	0.12	0.12	0.02	0.01	0.57	0.03
Love	0.13	0.01	0.08	0.14	0.01	0.00	0.21	0.08	0.34
Optimism	0.03	0.01	0.19	0.26	0.12	0.07	0.02	0.27	0.02
Pessimism	0.34	0.06	0.03	0.03	0.02	0.04	0.35	0.01	0.13
Sadness	0.01	0.02	0.01	0.01	0.00	0.01	0.03	0.00	0.91
Surprise	0.14	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.00
Trust	0.34	0.00	0.00	0.00	0.00	0.00	0.65	0.00	0.00

Figure 7. Attention visualization example. Golden labels are {*sadness*, *surprise*} and predicted labels are {*love (0.74)*, *sadness (0.98)*}.



	<user>	happy	birthday	gorg	,	have	a	good	one	😘	x
Anger	0.05	0.00	0.00	0.01	0.27	0.33	0.21	0.00	0.09	0.00	0.03
Anticipation	0.49	0.01	0.04	0.01	0.05	0.14	0.06	0.03	0.07	0.04	0.07
Disgust	0.04	0.00	0.00	0.01	0.34	0.32	0.16	0.00	0.10	0.00	0.03
Fear	0.60	0.00	0.00	0.00	0.07	0.03	0.28	0.00	0.01	0.00	0.01
Joy	0.01	0.67	0.08	0.09	0.00	0.00	0.00	0.09	0.00	0.06	0.00
Love	0.02	0.01	0.73	0.04	0.00	0.00	0.00	0.00	0.00	0.20	0.00
Optimism	0.10	0.29	0.09	0.04	0.00	0.00	0.01	0.33	0.01	0.11	0.01
Pessimism	0.40	0.00	0.00	0.00	0.34	0.05	0.17	0.01	0.01	0.01	0.02
Sadness	0.16	0.01	0.03	0.02	0.42	0.16	0.13	0.01	0.02	0.01	0.02
Surprise	0.98	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
Trust	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

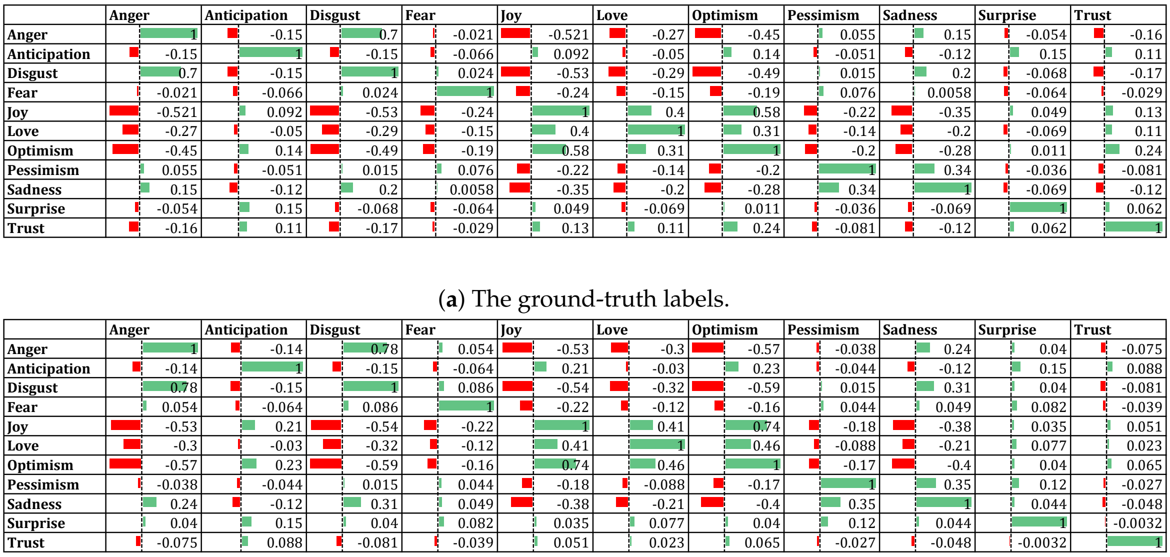
Figure 8. Attention visualization example. Golden labels are {*joy*, *love*, *optimism*} and predicted labels are {*joy (0.98)*, *love (0.91) optimism (0.95)*}.

4.7. Correlation Analysis

Figure 9 shows the correlation analysis of emotion labels in the validation set. Each cell in the figure represents the correlation score of each pair of emotion labels. The reported values show exciting findings. Our system captured the relations among the emotion labels. The correlation scores of the predicted labels were almost identical to the ground-truth. There was an exception in the surprise and trust emotions. Our system was unsuccessful in capturing the relationships between these two emotions and the inputs or the other emotions. We attribute this apparent lack of correlation to the low number of training examples of these two emotions.

Moreover, there was always a positive correlation between related emotions such as “joy” and “optimism” (the score from the ground truth labels and from the predicted labels was 0.74). On the other side, we can see that there was a negative correlation between unlinked emotions like “anger” and “love”. The scores were −0.27 and −0.3, respectively.

This result further strengthened our hypothesis that the proposed system was able to, implicitly, model the relationships between the emotion labels.



	Anger	Anticipation	Disgust	Fear	Joy	Love	Optimism	Pessimism	Sadness	Surprise	Trust
Anger	1	-0.15	0.7	-0.021	-0.521	-0.27	-0.45	0.055	0.15	-0.054	-0.16
Anticipation	-0.15	1	-0.15	-0.066	0.092	-0.05	0.14	-0.051	-0.12	0.15	0.11
Disgust	0.7	-0.15	1	0.024	-0.53	-0.29	-0.49	0.015	0.2	-0.068	-0.17
Fear	-0.021	-0.066	0.024	1	-0.24	-0.15	-0.19	0.076	0.0058	-0.064	-0.029
Joy	-0.521	0.092	-0.53	-0.24	1	0.4	0.58	-0.22	-0.35	0.049	0.13
Love	-0.27	-0.05	-0.29	-0.15	0.4	1	0.31	-0.14	-0.2	-0.069	0.11
Optimism	-0.45	0.14	-0.49	-0.19	0.58	0.31	1	-0.2	-0.28	0.011	0.24
Pessimism	0.055	-0.051	0.015	0.076	-0.22	-0.14	-0.2	1	0.34	-0.036	-0.081
Sadness	0.15	-0.12	0.2	0.0058	-0.35	-0.2	-0.28	0.34	1	-0.069	-0.12
Surprise	-0.054	0.15	-0.068	-0.064	0.049	-0.069	0.011	-0.036	-0.069	1	0.062
Trust	-0.16	0.11	-0.17	-0.029	0.13	0.11	0.24	-0.081	-0.12	0.062	1

(**a**) The ground-truth labels.

	Anger	Anticipation	Disgust	Fear	Joy	Love	Optimism	Pessimism	Sadness	Surprise	Trust
Anger	1	-0.14	0.78	0.054	-0.53	-0.3	-0.57	-0.038	0.24	0.04	-0.075
Anticipation	-0.14	1	-0.15	-0.064	0.21	-0.03	0.23	-0.044	-0.12	0.15	0.088
Disgust	0.78	-0.15	1	0.086	-0.54	-0.32	-0.59	0.015	0.31	0.04	-0.081
Fear	0.054	-0.064	0.086	1	-0.22	-0.12	-0.16	0.044	0.049	0.082	-0.039
Joy	-0.53	0.21	-0.54	-0.22	1	0.41	0.74	-0.18	-0.38	0.035	0.051
Love	-0.3	-0.03	-0.32	-0.12	0.41	1	0.46	-0.088	-0.21	0.077	0.023
Optimism	-0.57	0.23	-0.59	-0.16	0.74	0.46	1	-0.17	-0.4	0.04	0.065
Pessimism	-0.038	-0.044	0.015	0.044	-0.18	-0.088	-0.17	1	0.35	0.12	-0.027
Sadness	0.24	-0.12	0.31	0.049	-0.38	-0.21	-0.4	0.35	1	0.044	-0.048
Surprise	0.04	0.15	0.04	0.082	0.035	0.077	0.04	0.12	0.044	1	-0.0032
Trust	-0.075	0.088	-0.081	-0.039	0.051	0.023	0.065	-0.027	-0.048	-0.0032	1

(**b**) The predicted labels.

Figure 9. Correlation matrices of emotion labels of the development set.

5. Conclusions

In this work, we presented a new approach to the multi-label emotion classification task. First, we proposed a transformation method to transform the problem into a single binary classification problem. Afterwards, we developed a deep learning-based system to solve the transformed problem. The key component of our system was the embedding module, which used three embedding models and an attention function. Our system outperformed the state-of-the-art systems, achieving a Jaccard (i.e., multi-label accuracy) score of 0.59 on the challenging SemEval2018 Task 1:E-c multi-label emotion classification problem.

We found that the attention function can model the relationships between the input words and the labels, which helps to improve the system's performance. Moreover, we showed that our system is interpretable by visualizing the attention weights and analyzing them. However, some limitations have been identified. Our system does not model the relationships between the phrases and the labels. Phrases play a key role in determining the most appropriate set of emotions that must be assigned to a tweet. For instance, an emotion word that reflects "sadness" can be flipped in a negated phrase or context. Thus, in our future work, we plan to work on solving this drawback. One possible solution is to adapt the attention function to model the relationships between different *n*-gram tokens and labels. Structured attention networks [40] can also be adapted and used to address this issue.

Moreover, we plan to work on developing a non-aggressive system that performs robustly and equally on all the emotion labels by experimenting with different ideas like using data augmentation to enrich the training data or using transfer learning.

Author Contributions: Conceptualization, M.J. and A.M.; methodology, M.J.; software, M.J.; validation, M.J. and A.M.; formal analysis, M.J. and A.M.; investigation, M.J.; resources, M.J.; data curation, M.J.; writing, original draft preparation, M.J.; writing, review and editing, A.M.; visualization, M.J.; supervision, A.M.; funding acquisition, A.M.

Acknowledgments: The authors acknowledge the research support of URV(2018PFR-URV-B2 and Martí i Franqués PhD grant).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Zhou, Y.; Zhu, S.; Xu, H. Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. In Proceedings of the 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT), and 2012 International Conference on Social Computing (SocialCom), Amsterdam, The Netherlands, 3–5 September 2012; pp. 71–80.
2. Cherry, C.; Mohammad, S.M.; Bruijn, B. Binary Classifiers and Latent Sequence Models for Emotion Detection in Suicide Notes. *Biomed. Inform. Insights* **2012**, *5*, BII-S8933. [CrossRef] [PubMed]
3. Mohammad, S.M.; Zhu, X.; Kiritchenko, S.; Martin, J. Sentiment, Emotion, Purpose, and Style in Electoral Tweets. *Inf. Process. Manag.* **2015**, *51*, 480–499. [CrossRef]
4. Cambria, E. Affective computing and sentiment analysis. *IEEE Intell. Syst.* **2016** *31*, 102–107. [CrossRef]
5. Jabreel, M.; Moreno, A.; Huertas, A. Do Local Residents and Visitors Express the Same Sentiments on Destinations Through Social Media? In *Information and Communication Technologies in Tourism*; Springer: New York, NY, USA, 2017; pp. 655–668.
6. Yun, H.Y.; Lin, P.H.; Lin, R. Emotional Product Design and Perceived Brand Emotion. *Int. J. Adv. Psychol. IJAP* **2014**, *3*, 59–66. [CrossRef]
7. Mohammad, S.M. Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text. In *Emotion Measurement*; Meiselman, H.L., Ed.; Woodhead Publishing: Cambridge, UK, 2016; pp. 201–237, ISBN 9780081005088.
8. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier Chains for Multi-label Classification. *Mach. Learn.* **2011**, *85*, 333. [CrossRef]
9. Robert, P. Emotions: A general Psychoevolutionary Theory. In *Approaches to Emotion*; Scherer, K.R., Ekman, P., Eds.; Psychology Press: London, UK, 2014; pp. 197–219.

10. Tsoumakas, G.; Katakis, I. Multi-label Classification: An Overview. *Int. J. Data Warehous. Min. IJDWM* **2007**, *3*, 1–13. [CrossRef]
11. Read, J. Scalable Multi-label Classification. Ph.D. Thesis, University of Waikato, Hamilton, New Zealand, 2010.
12. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
13. Schuster, M.; Paliwal, K.K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
14. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
15. Al-Molegi, A.; Jabreel, M.; Martínez-Ballesté, A. Move, Attend and Predict: An attention-based neural model for people's movement prediction. *Pattern Recognit. Lett.* **2018**, *112*, 34–40. [CrossRef]
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
17. Zhang, M.L.; Zhou, Z.H. A review on Multi-label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1819–1837. [CrossRef]
18. Zhang, M.L.; Zhang, K. Multi-label Learning by Exploiting Label Dependency. In Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC, USA, 25–28 July 2010; pp. 999–1008.
19. Schapire, R.E.; Singer, Y. BoosTexter: A boosting-Based System for Text Categorization. *Mach. Learn.* **2000** *39*, 135–168. [CrossRef]
20. Zhang, M.L.; Zhou, Z.H. ML-KNN: A Lazy Learning Approach to Multi-label Learning. *Pattern Recognit.* **2007**, *40*, 2038–2048. [CrossRef]
21. Clare, A.; King, R.D. Knowledge Discovery in Multi-label Phenotype Data. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Freiburg, Germany, 3–5 September 2001; pp. 42–53.
22. De Comite, F., Gilleron, R.; Tommasi, M. Learning Multi-label Alternating Decision Trees from Texts and Data. In Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany, 5–7 July 2003; Volume 2734, pp. 35–49.
23. Mencia, E.L.; Fürnkranz, J. Efficient Pairwise Multilabel Classification for Large-Scale Problems in the Legal Domain. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, Belgium, 15–19 September 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 50–65.
24. Cheng, W.; Hüllermeier, E. Combining Instance-Based Learning and Logistic Regression for Multilabel Classification. *Mach. Learn.* **2009**, *76*, 211–225. [CrossRef]
25. Godbole, S.; Sarawagi, S. Discriminative Methods for Multi-labeled Classification. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Osaka, Japan, 20–23 May 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 22–30.
26. Younes, Z.; Abdallah, F.; Denoeux, T.; Snoussi, H. A Dependent Multilabel Classification Method Derived From the k-Nearest Neighbor Rule. *J. Adv. Signal Process.* **2011**, *1*, 645964. [CrossRef]
27. Yan, R.; Tesic, J.; Smith, J.R. Model-Shared Subspace Boosting for Multi-label Classification. In Proceedings of the 13th ACM SIGKDD, San Jose, CA, USA, 12–15 August 2007; pp. 834–843.
28. Jabreel, M.; Moreno, A. SentiRich: Sentiment Analysis of Tweets Based on a Rich Set of Features. In *Artificial Intelligence Research and Development*; Nebot, Á., Binefa, X., López de Mántaras, R., Eds.; IOS Press: Amsterdam, The Netherlands, 2016; Volume 288, pp. 137–146, ISBN 978-1-61499-695-8.
29. Jabreel, M.; Moreno, A. SiTAKA at SemEval-2017 Task 4: Sentiment Analysis in Twitter Based on a Rich Set of Features. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 694–699.
30. Mohammed, S., M.; Bravo-Marquez, F.; Salameh, M.; Kiritchenko,S. Semeval-2018 task 1: Affect in Tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 1–17.
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015** *521*, 436–444. [CrossRef]

32. Tang, D.; Qin, B.; Liu, T. Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015** *5*, 292–303. [CrossRef]
33. Baziotis, C.; Athanasiou, N.; Chronopoulou, A.; Kolovou, A.; Paraskevopoulos, G.; Ellinas, N.; Narayanan, S.; Potamianos, A. NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content in Tweets with Deep Attentive RNNs and Transfer Learning. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 245–255.
34. Meisheri, H.; Dey, L. TCS Research at Semeval2018 Task 1: Learning Robust Representations using Multi-Attention Architecture. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 291–299.
35. Park, J.H.; Xu, P.; Fung, P. PlusEmo2Vec at SemEval-2018 Task 1: Exploiting Emotion Knowledge from Emoji and #hashtags. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 264–272.
36. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
37. James, B.; Yamins, D.; Cox, D.D. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In Proceedings of the 12th Python in Science Conference, Austin, TX, USA, 24–29 June 2013.
38. Mohammad, S.; Kiritchenko, S. Understanding emotions: A dataset of tweets to study interactions between affect categories. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
39. Kant, N.; Puri, R.; Yakovenko, N.; Catanzaro, B. Practical Text Classification With Large Pre-Trained Language Models. *arXiv* **2018**, arXiv:1812.01207.
40. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured Attention Networks. *arXiv* **2017**, arXiv:1702.00887.