Multi-Label Sentiment Analysis on 100 Languages With Dynamic Weighting for Label Imbalance

Selim F. Yilmaz, *Graduate Student Member, IEEE*, E. Batuhan Kaynak, Aykut Koç, *Senior Member, IEEE*, Hamdi Dibeklioğlu, *Member, IEEE*, and Suleyman Serdar Kozat, *Senior Member, IEEE*

***Abstract*— We investigate cross-lingual sentiment analysis, which has attracted significant attention due to its applications in various areas including market research, politics, and social sciences. In particular, we introduce a sentiment analysis framework in multi-label setting as it obeys Plutchik's wheel of emotions. We introduce a novel dynamic weighting method that balances the contribution from each class during training, unlike previous static weighting methods that assign non-changing weights based on their class frequency. Moreover, we adapt the focal loss that favors harder instances from single-label object recognition literature to our multi-label setting. Furthermore, we derive a method to choose optimal class-specific thresholds that maximize the macro-f1 score in linear time complexity. Through an extensive set of experiments, we show that our method obtains the state-of-the-art performance in seven of nine metrics in three different languages using a single model compared with the common baselines and the best performing methods in the SemEval competition. We publicly share our code for our model, which can perform sentiment analysis in 100 languages, to facilitate further research.**

***Index Terms*— Cross-lingual, label imbalance, macro-f1 maximization, multi-label, natural language processing (NLP), sentiment analysis, social media.**

I. Introduction

A. Preliminaries

We study sentiment analysis problem in multi-label setting, which has been widely studied in the literature due to its significance in various applications including market research, politics, public health, and disaster management [1]–[3]. In particular, we introduce a method for cross-lingual sentiment analysis, which is a harder problem than the standard sentiment analysis problem since one needs to make predictions for various languages, including even unseen ones. Cross-lingual sentiment analysis aims to leverage high-quality and abundant resources in English for classification to improve the classification performance of resource-scarce languages [4]. Moreover, we use data of three languages to obtain the best score in seven of nine metrics of Arabic, English, and Spanish languages in the SemEval emotion classification [1].

Emotions are an integral part of human communication and decision-making mechanisms [5]. Plutchik [6], in 1980, has created the wheel of emotions in his psychoevolutionary theory of emotion to illustrate his idea of emotion. He suggests eight bipolar primary emotions that appear on the opposite sides of the wheel: joy versus sadness, anger versus fear, disgust versus trust, and surprise versus anticipation. The primary emotions are expressed at different intensities and the intermediate emotions occur as a mix of these primary emotions. Moreover, the emotions are non-exclusive in Plutchik's model as their combinations derive other emotions. There exist correlations between the emotions, for example, joy and sadness are represented as the opposite emotions. The Hourglass of Emotions [7], [8] reinterprets Plutchik's model by reorganizing the primary emotions along four independent dimensions. Following these emotion models, we formulate the sentiment analysis as the multi-label classification task, in which more than one label can be assigned to a text simultaneously. Yet, the class imbalance is an inherent issue in multi-label classification [9]. Although class imbalance has been extensively studied for the binary classification setting, it remains a challenge in multi-label classification [9]. Furthermore, the tail labels, that is, the labels with a low number of instances, impact the performance significantly less compared with the common labels when the classes are equally weighted in the multi-label setting due to the rarity of relevant examples and result in suboptimal performance [10]. Thus, we introduce a dynamic weighting method to dynamically adjust the class weights during training to remedy the class imbalance.

In this article, we introduce a multilingual sentiment analysis framework in multi-label setting on 100 different languages. Our method uses focal loss to enhance the importance of hard examples. We introduce a dynamic weighting

Manuscript received 26 August 2020; revised 8 April 2021; accepted 25 June 2021. Date of publication 19 July 2021; date of current version 5 January 2023. This work was supported in part by the Vodafone Turkey Within the Framework of 5G and Beyond Joint Graduate Support Programme Coordinated by Information and Communication Technologies Authority. *(Corresponding author: Selim F. Yilmaz.)*

Selim F. Yilmaz is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: syilmaz@ee.bilkent.edu.tr).

E. Batuhan Kaynak and Hamdi Dibeklioğlu are with the Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: batuhan.kaynak@bilkent.edu.tr; dibeklioglu@cs.bilkent.edu.tr).

Aykut Koç is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey, and also with the National Magnetic Resonance Research Center (UMRAM), Bilkent University, 06800 Ankara, Turkey (e-mail: aykut.koc@bilkent.edu.tr).

Suleyman Serdar Kozat is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey, and also with DataBoss A.S., 06800 Ankara, Turkey (e-mail: kozat@ee.bilkent.edu.tr; serdar.kozat@data-boss.com.tr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3094304.

Digital Object Identifier 10.1109/TNNLS.2021.3094304

method to cope with the label imbalance. We also derive a macro-f1 maximization method within linear time complexity. Our method achieves the best result for seven of nine metrics for the SemEval competition for Arabic, English, and Spanish languages [1]. We also demonstrate the performance of our method on cross-lingual combinations of the datasets and assess the performance gains obtained by the components in our method.

B. Prior Art and Comparisons

The current sentiment analysis methods can be represented under three different categories: knowledge-based (symbolic), statistical (sub-symbolic), and hybrid [11]. Knowledge-based methods mainly use manually or semi-automatically constructed lexicons to classify texts into affect categories [11], [12]. The knowledge-based methods are widely adopted because of their accessibility and affordability [11]. Single-language sentiment analysis usually uses lexicons or ontologies, which are manually or semi-automatically constructed [12]. SenticNet is a knowledge base of sentimental concepts, aiming to improve the generalization and interpretability of the sentiment analysis methods. Statistical approaches for sentiment analysis task include deep-learning-based methods, which we describe in the next paragraph. The hybrid methods adopt both knowledge-based approaches and statistical methods. SenticNet 6 [13] ensembles both symbolic and sub-symbolic approaches by integrating logical reasoning into deep learning framework.

The deep-learning-based methods have been shown to be successful in various classification tasks [14], [15]. SenticNet 5 [16] shows that concept clusters, which share similar syntactic or semantic functions, can automatically be discovered through a deep learning framework. Transfer learning approaches have been popular in sentiment analysis and shown to be successful, especially in datasets with a small number of instances [17]. Through transfer learning, large unlabeled corpora in social media have been incorporated to increase the target sentiment analysis task's performance, for example, [17] uses 1.7 billion tweets with emojis to pretrain the network. However, [18] demonstrates that the transfer learning approach does not improve the performance on the SemEval emotion classification competition datasets, which is our target due to the richness of its labels, which has significantly more number of instances compared with the number of instances used in [17]. Suttles and Ide [19] emulate a multi-label classifier through a binary classifier for each of the four opposite emotions that are on the opposite sides of Plutchik's wheel of emotions, such as joy and sadness. However, their approach does not include the correlations to the rest of the labels since they train each of the four classifiers with the objective of binary classification of the opposite sides. To remedy those issues, we introduce a multi-label deep learning model for the emoji prediction task that directly predicts the active set of labels simultaneously, that is, in the multi-task setting. Moreover, the multi-label classification is a generalization of binary and multi-class classification tasks as we describe through remarks. Thus, our method is also applicable to these tasks. Multi-label classification also requires a prediction method that converts scores into predictions, for which we derive a class-specific thresholding method by macro-f1 maximization in linear time complexity.

Multilingual sentiment analysis on social media has become an important problem as it is becoming mainstream media for communication and expression of thoughts in many different languages [12]. Being easily applicable to other languages, [20] leverages hashtags to collect sentiment classification dataset and performs training on it. Balahur and Turchi [21] translates an annotated English dataset to build classifiers in other languages. However, this method is prone to errors caused by translation. BabelSenticNet uses statistical machine translation to provide a concept-level multilingual sentiment analysis framework that applies to a wide range of domains [22]. Our method leverages XLM-RoBERTa [23] to perform cross-lingual and multilingual sentiment classification on tweets.

Multi-label classification has an inherent issue of data imbalance [9]. Although significant research has been performed in the literature, the class imbalance problem remains a challenge for multi-label classification [9]. Consider the multi-label classification task with 16 distinct labels. There are 2^{16} possible combinations in the superset of the labels. Accordingly, it is not feasible to obtain balanced data for each combination of the labels. Many studies in multi-output classification either try to balance the data by resampling or ignore the imbalance [9]. Yet, the over-sampling and under-sampling methods are not designed for multi-label classification; thus, their adaptation to the multi-label setting is not straightforward [9]. One heuristic that is widely adapted uses inverse class frequency per class as a weighting factor [24]. However, this heuristic results in suboptimal performance as shown by [25] and in Section IV-E. Cui *et al.* [25] replace the inverse number of instances with the expected volume of instances and a controlling hyperparameter. Aurelio *et al.* [26] propose to use class prior probabilities as weights for the cross-entropy loss. Commonly, these methods propose static weights for each class. To remedy the label imbalance in the multi-label setting, we introduce a novel dynamic weighting method, which equalizes the contribution of each class to the loss. We use focal loss [27] to incorporate the hardness of the instances and our dynamic weighting method can readily be adapted to other losses as we show through a remark.

Recent language models such as bidirectional encoder representations from transformers (BERT) [28] have been dominating the areas in the natural language processing (NLP) literature; however, they contain an excessive amount of parameters. Accordingly, training or fine-tuning these models require an excessive amount of resources [28]. We use XLM-RoBERTa [23], which is a robustly trained BERT on 100 languages, as feature extractor to benefit from BERT and reducing the number of required resources.

SemEval emotion classification competition [1] has paved the way for many multi-label sentiment analysis models. Emotion mining for Arabic (EMA) and PARTNA are among the models that opt for the more traditional support vector machine approaches and still achieve the best results in the Arabic

language [29]. On the other hand, more recent long short-term memory (LSTM), convolutional neural network (CNN), and attention models are also adopted to obtain the highest ranked results in English and Spanish [30], [31]. It is important to note that most of these models are language-specific and use special embeddings such as AraVec [32] or special lexicons paired with language-specific preprocessing steps. Tw-StAR attempts to create a generic model to apply multiple languages, yet is ranked behind the language-specific models [33]. We introduce a framework that uses bidirectional LSTM with attention and multi-label focal loss, which achieves the best score only using a single model on seven of the nine metrics on three different languages of the SemEval emotion classification competition [1].

C. Contributions

Our contributions are as follows.

1) For the first time in the literature to the best of our knowledge, we introduce a multi-label emotion classification method capable of producing uniformly high classification performance on 100 different languages using a single model. Our method can readily be adapted to the cross-lingual platforms such as Amazon without using any language detection component. We make our model publicly available[1] to facilitate reproducibility and further research.
2) We introduce a dynamic weighting method to remedy the class imbalance that is an inherent problem in multi-label classification with adaptive loss weights as training progress, unlike the previous static weighting methods [25], [26]. We demonstrate significant performance gains compared with the previous weighting methods. Our method performs no worse than the uniform weighting, that is, no weighting, for none of the hyperparameter choices. Our dynamic weighting method can be readily extended to other losses as we show through a remark.
3) We derive a method to maximize macro-f1 with class-specific threshold choices, which reduces the time complexity from exponential to linear. Moreover, we adapt focal loss to our multi-label emotion classification framework from the single-label object recognition literature, and we show performance improvements obtained via the focal loss [27].
4) Through an extensive set of experiments, we show that our model achieves the best scores in seven of nine metrics in the SemEval emotion classification competition for Arabic, English, and Spanish via a single model [1]. Furthermore, we perform cross-lingual experiments, hyperparameter studies, and an ablation study to assess the effectiveness of our method.

D. Organization of This Article

The rest of the article is organized as follows. In Section II, we describe the multi-label sentiment analysis task and show that it is the generalization of the binary and multi-class classification tasks. In Section III, we introduce our deep metric learning-based framework and the components to cope with the label imbalance. In Section IV, we demonstrate the performance improvements obtained by our proposed model compared with the state-of-the-art methods in the literature and the SemEval [1] emotion classification competition winners. In Section V, we conclude by providing remarks.

[1]https://github.com/selimfirat/multilingual-sentiment-analysis

II. Problem Description

In this article, all vectors are column vectors and defined by boldfaced lowercase letters. All matrices and tensors are represented by boldfaced uppercase letters. $|\cdot|$ denotes cardinality, that is, the number of elements, of set $\cdot$.

We aim to predict the labels of the given text in multi-label framework through our network $\mathcal{F}$. We receive training data $\mathcal{P} = \{(s_i, \boldsymbol{c}_i)\}_i^n$, where s_i is the text of ith training instance, n is the number of training instances, $\boldsymbol{c}_i = [c_{i,1}\ c_{i,2}, \ldots, c_{i,w}]^T$ is the label vector of the ith training instance, w is the number of classes, and $c_{i,a}$, $a \in \{1, 2, \ldots, w\}$, is defined by

$$c_{i,a} = \begin{cases} 1, & \text{if class } a \text{ is inferred} \\ 0, & \text{otherwise.} \end{cases}$$

To satisfy this decision function, we predict score $\hat{c}_{i,a}$ for the target sentence s_i via our network $\mathcal{F}$ as

$$\hat{c}_{i,a} = \mathcal{F}(s_i) = p(c_{i,a} = 1 \mid s_i). \tag{1}$$

Remark 1: Multi-class classification is a generalization of multi-class and binary classification tasks. For both, we have only one active label, that is, $\sum_{a=1}^{w} c_{i,a} = 1,\ \forall i \in \{1, 2, \ldots, n\}$. The number of classes $w = 2$ and $w > 2$ for binary and multi-label classification, respectively. Since we formulate the problem as a multi-label classification, our framework is applicable to binary classification, multi-class classification, and multi-label classification settings.

III. Methodology

In this section, we first describe language modeling and recurrent modeling with attention for multi-label classification. We then introduce our multi-label adaptation of focal loss and our dynamic weighting method. Finally, we derive a method to select thresholds by maximizing macro-f1 within linear time complexity. Fig. 1 illustrates the overall structure of our methodology.

A. Deep Multilingual Language Modeling

Here, we describe our language modeling approach using XLM-RoBERTa [23].

Traditional approaches such as well-known Bag-of-Words fail to generalize to unseen data due to the sparsity of the language [34]. Early word embedding-based methods, such as the well-known word2vec [35], based approaches have been used to cope with this problem via learning a vector for each word in a large vocabulary exploiting semantic relationships between words [34]. However, these methods assign a single vector to each word regardless of the context

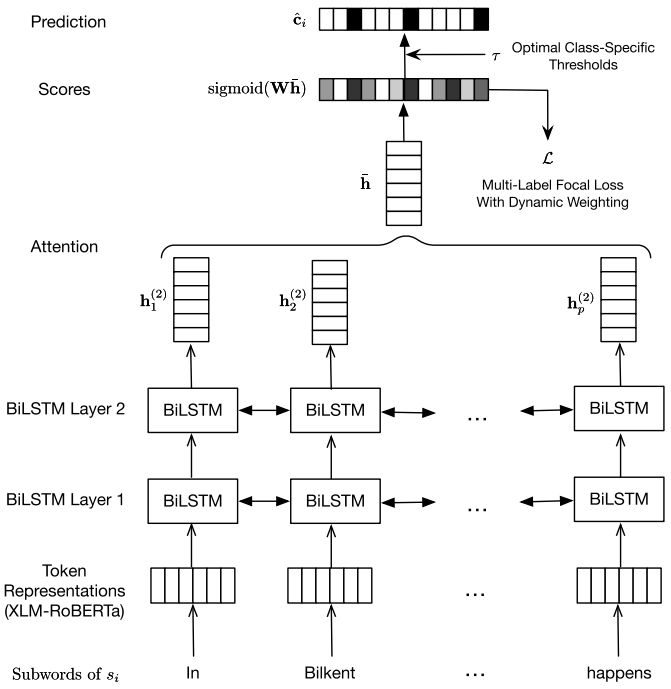

Fig. 1. Overall structure of our model.

of the target sentence. Recently, language models such as BERT have achieved outstanding results on various tasks [28]. These language models assign context-dependent vectors to each token in the target space instead of assigning a fixed vector. These models are trained using large corpora in an unsupervised setting. However, these models contain millions of parameters, and it is not reasonable to fine-tune them on a small corpus. Thus, we use feature vectors extracted from the pretrained model for each text instead of directly fine-tuning the pretrained model.

As shown in Fig. 1, we use XLM-RoBERTa pretrained tokenizer and pretrained model [23]. XLM-RoBERTa is pretrained on CommonCrawl corpora of 100 different languages. We first tokenize the input sentence s_i into subword units via byte-pair encoding using Sentencepiece tokenizer [36]. We convert the sentence s_i to $\boldsymbol{X}^{(i)} \in \mathbb{R}^{m \times d_i}$ using the hidden state vectors of the pretrained language model, where m is the embedding vector length and d_i is the number of tokens in the sentence s_i. We obtain an embedding vector for each token in the sentence, that is, $\boldsymbol{X}^{(i)} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d_i}]$, where $\boldsymbol{x}_j \in \mathbb{R}^m, \forall j \in \{1, 2, \ldots, d_i\}$.

Remark 2: Our model can be adapted to other languages since we tokenize via byte-pair encoding and convert to features without applying any language-dependent preprocessing. For the languages that XLM-RoBERTa does not support, one can directly use any other pretrained model that supports the target language. We show the cross-lingual performance of our method in Section IV-D.

B. Temporal Modeling of Sentence via Recurrent Networks

Here, we describe our recurrent modeling for multi-label emotion classification using the frozen features via the language modeling network.

We are given a sequence of token embeddings $\boldsymbol{X}^{(i)} \in d_i \times m$ for the sentence s_i, where d_i is the number of tokens in the sentence s_i and m is the embedding size. $\boldsymbol{x}_k \in \mathbb{R}^m$ indicates the embedding of the kth token.

As shown in Fig. 1, we use bidirectional RNNs to incorporate both the forward and the backward information of the sequence. Through RNN, we process the variable length sequences. We use deep networks, where the number of layers is u. For timestep t and kth layer, we use $\overrightarrow{\boldsymbol{h}}_t^{(k)}$ and $\overleftarrow{\boldsymbol{h}}_t^{(k)}$ notations to define forward and backward RNNs, respectively. We define kth layer of the forward RNN that uses Elman's formulation [37] as

$$\overrightarrow{\boldsymbol{h}}_t^{(k)} = \tanh\left(\boldsymbol{W}_{\text{hh}}^{(k)} \overrightarrow{\boldsymbol{h}}_{t-1}^{(k)} + \boldsymbol{W}_{\text{hx}}^{(k)} \overrightarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{b}^{(k)}\right)$$

where $\overrightarrow{\boldsymbol{h}}_t^{(0)} = \boldsymbol{x}_t$ for $t \in \{1, 2, \ldots, d_i\}$ $\overrightarrow{\boldsymbol{h}}_0^{(k)} \sim \mathcal{N}(0, 0.01)$, $\boldsymbol{b}^{(k)}$ is the bias term to be learned, and $\boldsymbol{W}_{\text{hh}}^{(k)}$ and $\boldsymbol{W}_{\text{hx}}^{(k)}$ are the weights to be learned. We also define the backward RNN's hidden state $\overleftarrow{\boldsymbol{h}}_t^{(k)}$ for the kth layer by feeding the reversed input to the RNN, that is,

$$\overleftarrow{\boldsymbol{h}}_t^{(k)} = \tanh\left(\boldsymbol{V}_{\text{hh}}^{(k)} \overleftarrow{\boldsymbol{h}}_{t+1}^{(k)} + \boldsymbol{V}_{\text{hx}}^{(k)} \overleftarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{c}^{(k)}\right)$$

where $\overleftarrow{\boldsymbol{h}}_t^{(0)} = \boldsymbol{x}_{d_i - t + 1}$, $\overleftarrow{\boldsymbol{h}}_{d_i}^{(k)} \sim \mathcal{N}(0, 0.01)$, $\boldsymbol{c}^{(k)}$ is the bias term to be learned, and $\boldsymbol{V}_{\text{hh}}^{(k)}$ and $\boldsymbol{V}_{\text{hx}}^{(k)}$ are the weights to be learned.

Remark 3: We extend our framework to LSTM [38] due to its success in capturing complex temporal relationships. We feed the input sentence embedding $\boldsymbol{X}^{(i)}$ to the LSTM instead of the RNN as

$$\begin{aligned}
\boldsymbol{z}_t^{(k)} &= \tanh\left(\boldsymbol{W}_z^{(k)} \overrightarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{V}_z^{(k)} \overrightarrow{\boldsymbol{h}}_{t-1}^{(k)} + \boldsymbol{b}_z^{(k)}\right) \\
\boldsymbol{s}_t^{(k)} &= \text{sigmoid}\left(\boldsymbol{W}_s^{(k)} \overrightarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{V}_s^{(k)} \overrightarrow{\boldsymbol{h}}_{t-1}^{(k)} + \boldsymbol{b}_s^{(k)}\right) \\
\boldsymbol{f}_t^{(k)} &= \text{sigmoid}\left(\boldsymbol{W}_f^{(k)} \overrightarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{V}_f^{(k)} \overrightarrow{\boldsymbol{h}}_{t-1}^{(k)} + \boldsymbol{b}_f^{(k)}\right) \\
\boldsymbol{c}_t^{(k)} &= \boldsymbol{s}_t^{(k)} \odot \boldsymbol{z}_t^{(k)} + \boldsymbol{f}_t^{(k)} \odot \boldsymbol{c}_{t-1}^{(k)} \\
\boldsymbol{o}_t^{(k)} &= \text{sigmoid}\left(\boldsymbol{W}_o^{(k)} \overrightarrow{\boldsymbol{h}}_t^{(k-1)} + \boldsymbol{R}_o^{(k)} \overrightarrow{\boldsymbol{h}}_{t-1}^{(k)} + \boldsymbol{b}_o^{(k)}\right) \\
\overrightarrow{\boldsymbol{h}}_t^{(k)} &= \boldsymbol{o}_t^{(k)} \odot \tanh\left(\boldsymbol{c}_t^{(k)}\right)
\end{aligned}$$

where $\overrightarrow{\boldsymbol{h}}_t^{(0)} = \boldsymbol{x}_t$, $\overrightarrow{\boldsymbol{h}}_0^{(k)} \sim \mathcal{N}(0, 0.01)$, $\boldsymbol{c}_t^{(k)} \in \mathbb{R}^m$ is the cell state vector, and $\boldsymbol{h}_t^{(k)} \in \mathbb{R}^w$ is the hidden state vector, for the t^{th} LSTM unit. $\boldsymbol{s}_t^{(k)}$, $\boldsymbol{f}_t^{(k)}$, and $\boldsymbol{o}_t^{(k)}$ are the input, forget, and output gates, respectively. $\odot$ is the operation for elementwise multiplication. $\boldsymbol{W}$, $\boldsymbol{V}$, and $\boldsymbol{b}$ with the subscripts z, s, f, and o are the parameters of the LSTM unit to be learned. We also define the backward LSTM via $\overleftarrow{\boldsymbol{h}}_t^{(k)}$ by reversing the input order for each layer of the LSTM, as in RNNs.

We concatenate the hidden states of the backward and forward RNNs of the kth layer at time t as

$$\boldsymbol{h}_t^{(k)} = \begin{bmatrix} \overrightarrow{\boldsymbol{h}}_t^{(k)} \\ \overleftarrow{\boldsymbol{h}}_t^{(k)} \end{bmatrix}.$$

We then apply attention to the hidden states by weighing each timestep's hidden state with a single parameter as [39]

$$\bar{\boldsymbol{h}} = \sum_{t=1}^{p} \beta_t \boldsymbol{h}_t^{(u)}$$

where p is the sequence length and $\beta_t = (\exp(\boldsymbol{h}_t \boldsymbol{s}))/(\sum_{i=1}^{p} \exp(\boldsymbol{h}_i \boldsymbol{s}))$ for the timestep $t \in \{1, 2, \ldots, p\}$. Finally, we use linear layer and sigmoid activation to convert our predictions to the labels as

$$\boldsymbol{r} = \text{sigmoid}(\boldsymbol{W}\bar{\boldsymbol{h}}) \tag{2}$$

where $\boldsymbol{r} \in \mathbb{R}^s$ and s is the number of the target labels of the task.

Remark 4: We use sigmoid activation at the final layer instead of softmax since the softmax assumes independence between labels, whereas in our case, the labels are non-independent due to Plutchik's theory [6] as we describe in Section I-A.

C. Multi-Label Focal Loss

In this section, we adapt the focal loss for our multi-label framework from the single-label object recognition literature [27]. We define $p_{i,a}$ for notational convenience as the following:

$$p_{i,a} = \begin{cases} r_{i,a}, & \text{if } c_{i,a} = 1 \\ 1 - r_{i,a}, & \text{otherwise} \end{cases}$$

where $r_{i,a}$ is the sigmoid output for class a and the instance i, which is obtained via (2). Then, the standard cross-entropy loss for instance i and class a becomes $-\log p_{i,a}$.

Focal loss has been proposed to overcome the class imbalance problem in object recognition, which extends the cross-entropy loss [27]. The focal loss focuses training of the hard instances instead of the well-classified ones as

$$l_{i,a} = -(1 - p_{i,a})^{\gamma} \log p_{i,a}$$

where $\gamma \in \mathbb{R}$ is a tunable parameter and $\gamma \geq 0$. Note that the focal loss extends the cross-entropy loss by multiplying with $(1 - p_{i,a})^{\gamma}$.

We convert the loss into a scalar by taking weighted sum with respect to the classes and averaging with respect to the instances in the batch as

$$\mathcal{L} = \frac{1}{b} \sum_{i=1}^{b} \sum_{a=1}^{w} \alpha_{t,a} l_{i,a} \tag{3}$$

such that $\sum_{a=1}^{w} \alpha_{t,a} = 1$, t is the index of the mini-batch iteration, b is the batch size, and $\alpha_{t,a}$ is the weight of the class a at the mini-batch iteration t. We can assign equal weights to by setting $\alpha_{t,a} = 1/w$ for each class a and for all mini-batch iteration t. In Section III-D, we introduce a novel method for choosing $\alpha_{t,a}$ to remedy the class imbalance.

D. Novel Dynamic Weighting Method for Label Imbalance

Here, we introduce our dynamic weighting method to improve the imbalanced multi-label classification, which can also be applied to single-label problems and other loss functions, as we show through remarks.

Although focal loss improves the imbalanced classification performance, there is still plenty of room for improvement. For instance, [27] uses an alpha balanced variant of the focal loss in practice, where they choose the inverse frequency of the class as in the imbalanced classification. Cui *et al.* [25] also extend focal loss by class-volume-based formulation and introduces another hyperparameter. We introduce a method to equalize the losses from all classes in the problem. Our goal is to define weights in a way that each class has an equal contribution to the loss, that is,

$$\sum_{i=1}^{|\mathcal{P}|} \alpha_{t,1} l_{i,1} = \sum_{i=1}^{|\mathcal{P}|} \alpha_{t,2} l_{i,2} = \cdots = \sum_{i=1}^{|\mathcal{P}|} \alpha_{t,w} l_{i,w}$$

where $\mathcal{P}$ is the training data.

Finding the exact value for $\alpha_{t,a}$ is intractable since model parameters change after each mini-batch and we train in mini-batch setting. Thus, we track the losses by exponentially smoothed approximation $\omega_{t,a}$ at mini-batch iteration t and class a, which is given by

$$\omega_{t,a} = (1 - \kappa)\omega_{t-1,a} + \kappa \sum_{i=1}^{b} l_{i,a}$$

where κ is the smoothing hyperparameter to be tuned and $\omega_{1,a} = 1/w, \forall a \in \{1, 2, \ldots, w\}$. We invert $\omega_{t,a}$ and introduce a very small ϵ term if there appears no loss for any class for numerical stability of our method since we may get 0 loss for some classes, as the following:

$$\phi_{t,a} = \frac{1}{\epsilon + \omega_{t,a}}$$

where we set $\epsilon = 1 \times 10^{-5}$. Using $\phi_{t,a}$, we define $\alpha_{t,a}$ in (3) as

$$\alpha_{t,a} = \frac{\phi_{t,a}}{\sum_{u=1}^{w} \phi_{t,u}}. \tag{4}$$

Through (4), we guarantee that the weights sum up to 1 for any mini-batch iteration. We set the gradient with respect to the network parameters Θ to zero, that is, $\nabla_{\Theta} \alpha_{t,a} = 0, \forall t \in \{1, 2, \ldots\}, a \in \{1, 2, \ldots, w\}$. We balance the loss contribution from the classes using $\alpha_{t,a}$ in (3).

Remark 5: Note that our dynamic weighting method's weights change over time with respect to the hardness of the instances among classes, unlike the previous methods in the literature [25], [26].

Remark 6: Dynamic weighting is loss-agnostic and, thus, can readily be adapted to other alternative losses. For example, we can adapt it into the cross-entropy loss by setting $l_{i,a} = -\log p_{i,a}$ and directly use (3).

Remark 7: The dynamic weighting method can also be applied to the single-label problems without any change since the multi-label problem is a generalization of the single-label variant.

E. Class-Specific Thresholding via Macro-F1 Maximization

We derive a macro-f1 maximization method by choosing the optimal class-specific threshold within linear time complexity.

We have the model output $\hat{\boldsymbol{c}}_{\boldsymbol{i}} = \boldsymbol{r}_i$, which is our score vector, that is to be thresholded to make a prediction. We have a class-specific score $\hat{c}_{i,a}$ for class a. We expect high scores for the inferred classes and low scores for the non-inferred classes. We split a part of the validation set as the thresholding set and

then use it to choose the optimal threshold that maximizes the macro-f1 score. We concatenate the scores of all instances in the thresholding set $\mathcal{T}$ into $\hat{\boldsymbol{c}}_a \in \mathbb{R}^{|\mathcal{T}|}$ as

$$\hat{\boldsymbol{c}}_a = \begin{bmatrix} \hat{c}_{1,a} & \hat{c}_{2,a} & \cdots & \hat{c}_{|\mathcal{T}|,a} \end{bmatrix}^T.$$

Our aim is to find a threshold vector $\boldsymbol{\tau} \in \mathbb{R}^w$ given by

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_w \end{bmatrix}^T.$$

We select the optimal threshold for each class that maximizes the macro-f1 score, which is the F1 score calculated for each class and averaged among the classes. F1 is the harmonic mean of the precision and recall for a class a, that is,

$$\text{F1}(\boldsymbol{c}_a, \hat{\boldsymbol{c}_a}) = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

such that

$$\text{precision}(\boldsymbol{c}_a, \hat{\boldsymbol{c}_a}) = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{recall}(\boldsymbol{c}_a, \hat{\boldsymbol{c}_a}) = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively.

Definition 1: The $\arg\max_\rho f(\rho)$ function returns ρ that maximizes the proceeding function $f(\rho)$. Note that if more than one value of ρ maximizes the function $f(\rho)$, then $\arg\max_\rho f(\rho)$ returns the minimum ρ among the ones maximizing $f(\rho)$.

As shown in Fig. 1, we select the threshold vector $\boldsymbol{\tau}$ to threshold the model scores by maximizing the MacroF1 function on the validation set as

$$\boldsymbol{\tau} = \arg\max_{\boldsymbol{\tau}} \text{MacroF1}(\delta(\hat{\boldsymbol{c}_a} \geq \tau_a), \boldsymbol{c}_a) \tag{5}$$

where $\delta(\cdot)$ function returns the same-sized vector with its input, which outputs 1 for the dimensions that satisfy inequality and 0 for the rest. Directly optimizing (5) via grid search becomes infeasible as the number of classes increases and it may not be possible to find the optimal value since the time complexity is in $\Theta(T^w)$,[2] where T is the number of elements to be tried and w is the number of classes. By expanding the MacroF1 function in (5), we obtain

$$\boldsymbol{\tau} = \arg\max_{\boldsymbol{\tau}} \frac{1}{w} \sum_{a=1}^{w} \text{F1}(\delta(\hat{\boldsymbol{c}_a} \geq \tau_a), \boldsymbol{c}_a)$$
$$= \frac{1}{w} \sum_{a=1}^{w} \arg\max_{\boldsymbol{\tau}} \text{F1}(\delta(\hat{\boldsymbol{c}_a} \geq \tau_a), \boldsymbol{c}_a).$$

Since the class-specific thresholds in $\boldsymbol{\tau}$ are independent, we separate the thresholds into different arg max functions as

$$\tau_a = \arg\max_{\tau_a} \text{F1}(\delta(\hat{\boldsymbol{c}_a} \geq \tau_a), \boldsymbol{c}_a).$$

Thus, we obtain that $\boldsymbol{\tau}$ is equivalent to $[\tau_1\ \tau_2\ \cdots\ \tau_w]^T$ such that

$$\tau_a = \arg\max_{\tau_a} \text{F1}(\delta(\hat{\boldsymbol{c}_a} \geq \tau_a), \boldsymbol{c}_a) \tag{6}$$

for all $a \in \{1, 2, \ldots, w\}$.

[2] $\Theta(w(n))$ denotes the set of all $r(n)$, where $a_1 w(n) \leq r(n) \leq a_2 w(n)$, $\forall n > n_0$ for $n \in \mathbb{Z}^+$ such that there exist positive integers a_1, a_2, and n_0.

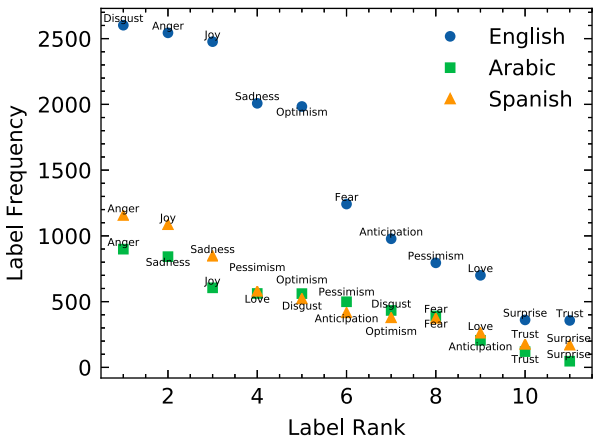

Fig. 2. Number of label occurrences versus rank plot of the labels in datasets that demonstrate the class imbalance.

Using (6), the time complexity becomes linear with respect to the number of classes, that is, $\Theta(Tw)$. Thus, we calculate the threshold vector $\boldsymbol{\tau}$ using (6).

Remark 8: Note that it is not reasonable to choose the threshold using the training set since the model already memorizes it and unavoidably performs biased scoring for the training data. This is why we use the unseen thresholding set.

IV. Experiments

In this section, we first describe the datasets, the evaluation methodology, and the implementation details. We then compare our method with the first ranking methods in the SemEval emotion classification competition [1] and the state-of-the-art methods. We then analyze the performance of our method via cross-lingual experiments. Later, we demonstrate performance gains obtained via our dynamic weighting method and analyze its hyperparameter. Finally, we present our method's individual class performances and demonstrate the performance gains obtained by our method's components via an ablation study.

A. Datasets

We use datasets in three different languages from the SemEval competition [1]: SemEval-Arabic, SemEval-English, and SemEval-Spanish. For simplicity, we refer SemEval-Arabic, SemEval-English, and SemEval-Spanish datasets as Arabic, English, and Spanish, respectively. Since the datasets are in multi-label setting, the instances contain zero or more labels among the 11 labels in the dataset. Fig. 2 demonstrates the class imbalance via the number of occurrence versus rank plot of the labels in the datasets. We use the splits of the SemEval emotion classification competition [1]. Arabic dataset has a total of 4381 instances consisting of 160 206 tokens and split into 3561 training, 679 validation, and 2854 test instances. The English dataset has 10 983 instances consisting of 338 763 tokens and split into 6838 training, 886 validation, and 3259 test instances. The Spanish dataset has 7094 instances consisting of 176 650 tokens and split into 2278 training, 585 validation, and 1518 test instances.

B. Evaluation Methodology and Implementation Details

We use macro averaged F1 (macro-f1), micro averaged F1 (micro-f1), and Jaccard index, which are the metrics used in the SemEval competition [1]. For fairness, we optimize our network, BERT [28], RoBERTa [43], XLM-RoBERTa [43], and deepmoji [17] baselines using Tree Parzen Estimator of the Optuna library [45] and choose the model with the largest validation macro-f1 score among 100 trials. For the FastText [40] baseline, we use its own hyperparameter optimization module with 130 different trials for each of the languages, that is, 30 more trials than our optimization for our model and the deepmoji baseline. For the methods in the SemEval emotion classification competition, we directly use their reported scores for fairness. These methods have also followed similar approaches for hyperparameter optimization, for example, EMA [29] performs a grid search, and NVIDIA [18] and NTUA-SLP [30] use Bayesian optimization in dimensional space of all the possible values.

We train our model via the Adam [46] optimizer. We use ekphrasis[3] preprocessing library to perform language-independent preprocessing of social cues such as username normalization. We use weight decay and early stopping. We stop the training when ten epochs are exceeded without any macro-f1 improvements on the validation set.

C. Comparison With the State-of-the-Art

Here, we compare our model with the state-of-the-art methods and the best models in SemEval-2018 competition in Arabic, English, and Spanish languages.

To create our best model, we combine the Arabic, English, and Spanish data by combining their training and validation sets. We then train our model on the combined data using the methodology described in Section IV-B.

We use 15 different baselines to compare our method and demonstrate its effectiveness, most of which are the highest performing contenders in SemEval-2018 emotion classification competition. NTUA-SLP [30] ranked first on Jaccard and micro-f1 metrics for English using a pretrained Bi-LSTM with a multi-layer self-attention mechanism. They use word2vec embeddings that are trained on 550 million tweets. The best micro-f1 score for English is achieved by psyML [31], which uses a very similar Bi-LSTM with self-attention model to NTUA-SLP, except they use hierarchical clustering to group correlated emotions together and train the same model incrementally for emotions within the same cluster. NVIDIA [18] trains an attention-based transformer network on large-scale data and fine-tunes this model on the training set for SemEval-English before testing it, obtaining results on par with those in the competition ranking. Deepmoji is a distant supervision-based LSTM architecture and it obtains the state-of-the-art performance on many sentiment-related tasks [17]. They convert multi-label instances into separate binary tasks. We report the results of their chain-thaw approach on the English dataset. For Arabic, EMA [29] (first place in Jaccard and micro-f1, second place in macro-f1) and PARTNA (first place in macro-f1, second place in Jaccard and micro-f1) achieve the highest two ranks. EMA uses AraVec embeddings [32] as features into a support vector classifier (SVC) with L_1 regularization. PARTNA uses a similar support-vector-based model except using an additional Arabic stemmer designed for handling tweets [1]. There are also studies that perform well but are not in SemEval rankings. Among these, context-aware gated recurrent unit (CA-GRU) [41] uses context information, the topic of the text in this case, as a feature by first feeding the text to a topic-detection model to obtain a vector of probability distributions over topics. HEF-DF [42] is a simple neural network hybrid model obtained from concatenating human-engineered (i.e., handpicking features that represent syntactical and semantical significance) and deep features (i.e., using combinations of embeddings). As with Arabic, two models exist for the first place in a metric for Spanish: MILAB_SNU (first place in Jaccard and micro-f1, second place in macro-f1) and ELiRF-UPV, which uses manually and automatically generated lexicon sand combines 1-D CNNs with an LSTM to obtain the first place in macro-f1 metric [44] (second place in Jaccard and micro-f1). We also include Tw-StAR [33] as a baseline to compare our method's multilingual performance with a standard model. Tw-StAR uses binary relevance transformation strategy to extract term frequency-inverse document frequency (tf-idf) features for a linear support vector machine. They also experiment with combinations of five different preprocessing methods and reach the third rank for both the Arabic and Spanish datasets. FastText is a framework that can convert text into feature vectors using a skip-gram model, where each word is represented as a bag of n-grams [40]. FastText contains readily extracted vectors for 157 languages. We fine-tune these vectors for English, Arabic, and Spanish and use them on their respective SemEval datasets. We use pretrained BERT [28], RoBERTa [43], and XLM-RoBERTa [43] as additional baselines, which are described in Section I-B, and fine-tune all the parameters of them.

Table I presents the results of our model compared with the state-of-the-art models and the competition winners. The models that target only a single language perform significantly better compared with the multilingual models. The only exception is our best model, which we train on three different languages' training data combined. Our best model obtains significantly better results compared with our single-language (Ours-SL) model with the same methodology and trained on each of these languages separately. Our method achieves the best score on all of the metrics in Arabic and Spanish languages. Our method achieves the best score in macro-f1 metric of the English language. In Arabic, our method achieves 4.8% (absolute) macro-f1 improvement compared with the previous best model, which is HEF-DF [42]. Our method obtains 2.3% (absolute) micro-f1 and 0.2% (absolute) Jaccard score improvement compared with the previous best model CA-GRU [41]. In English, our method achieves 1% (absolute) macro-f1 improvement compared with the previous competition winner psyML [31]. Our method attains 0.5% (absolute) micro-f1 and 1.2% (absolute) Jaccard score improvements compared with the competition winner NTUA-SLP [30]. Note that RoBERTa [43] model improves over NTUA-SLP

[3]https://github.com/cbaziotis/ekphrasis/tree/master/ekphrasis

TABLE I

COMPARISON OF OUR METHOD WITH THE STATE-OF-THE-ART METHODS IN THE LITERATURE AND THE WINNERS IN SEMEVAL-2018 EMOTION CLASSIFICATION COMPETITION [1] WITH THE INTRODUCED METHOD. THE PREVIOUSLY REPORTED STATE-OF-THE-ART RESULTS IN THE SEMEVAL COMPETITION ARE UNDERLINED. THE CURRENT STATE-OF-THE-ART RESULTS ARE BOLDFACED

	Arabic			English			Spanish		
Method	Macro-F1	Micro-F1	Jaccard	Macro-F1	Micro-F1	Jaccard	Macro-F1	Micro-F1	Jaccard
Ours	**55.0**	**66.1**	**53.4**	**58.4**	69.6	57.6	**53.0**	**60.6**	**48.6**
Ours-SL	51.3	57.5	44.3	56.4	68.7	56.5	50.5	56.6	45.3
Tw-StAR [33]	44.6	59.7	46.5	45.2	60.7	48.1	39.2	52.0	43.8
FastText [40]	35.3	40.2	25.5	35.0	39.9	25.5	27.0	31.9	20.6
XLM-RoBERTa [23]	48.2	65.2	53.0	56.1	69.3	57.3	48.3	57.0	47.3
CA-GRU [41]	49.5	64.8	53.2	-	-	-	-	-	-
HEF-DF [42]	50.2	63.1	51.2	-	-	-	-	-	-
EMA [29]	46.1	61.8	48.9	-	-	-	-	-	-
PARTNA	47.5	60.8	48.4	-	-	-	-	-	-
NTUA-SLP [30]	-	-	-	52.8	70.1	58.8	-	-	-
psyML [31]	-	-	-	57.4	69.7	57.4	-	-	-
NVIDIA [18]	-	-	-	56.1	69.0	57.7	-	-	-
DeepMoji [17]	-	-	-	55.9	65.7	52.8	-	-	-
BERT [28]	-	-	-	54.7	69.3	57.4	-	-	-
RoBERTa [43]	-	-	-	57.4	**71.0**	**59.1**	-	-	-
ELiRF-UPV [44]	-	-	-	-	-	-	44.0	53.5	45.8
MILAB_SNU	-	-	-	-	-	-	40.7	55.8	46.9

by 0.9% (absolute) macro-f1 and 0.3 (absolute) Jaccard score. In Spanish, our method achieves 9.0% (absolute) macro-f1 improvement compared with the previous best model ELiRF-UPV [44]. Our method also achieves 4.8% (absolute) micro-f1 and 1.7% (absolute) Jaccard score improvement compared with the previous best model MILAB_SNU.

We perform post hoc Nemenyi test among the multilingual models (Ours, Ours-SL, Tw-StAR, FastText, XLM-RoBERTa) [47]. Note that we could not perform post hoc Nemenyi test for all models here since not all of them report results on different datasets. We first perform Friedman test where we obtain $p < 0.05$ for each of macro-f1, micro-f1, and Jaccard index metrics. Then, we proceed with post hoc Nemenyi test. In the Nemenyi part, we have again repeated the test for each metric and obtained the same result from each: we can only reject the null hypothesis with $p < 0.05$ that results of the FastText and our best model were drawn from the same distribution. Note that since there are only three datasets, it is not possible to reject more than one in the optimal scenario since the maximum possible average rank difference is 4 and the critical difference is 3.52 (which must be greater than the average rank difference to reject).

D. Cross-Lingual Experiments

In this section, we demonstrate the cross-lingual capability of our method using training and test data combinations of different languages.

Table II presents the results when a model is trained on combinations of the datasets of different languages from the SemEval competition [1]. For each row, we train the model using the combined training data of the languages in the first column and validate using the combined validation data. We then experiment on the test sets of the English (EN), Spanish (SP), and Arabic (AR) languages separately. Note that the threshold and the best model are selected using the validation set of the combined data using the procedure we describe in Section IV-B. Recall that we use only a single model, which is the model shown in the last row (AR + EN + SP) in comparisons with the state-of-the-art in Section IV-C.

TABLE II

EXPERIMENT RESULTS WHEN THE MODEL IS TRAINED ON THE COMBINATIONS OF THE SEMEVAL DATASETS AND TESTED ON THE INDIVIDUAL VALIDATION SETS. THE BEST RESULTS ARE BOLDFACED

	Validation Data		
Training Data	Arabic	English	Spanish
Arabic (SP)	52.7	39.6	30.7
English (EN)	37.9	60.1	36.2
Spanish (SP)	35.2	46.4	52.3
AR + EN	52.5	61.1	39.6
AR + SP	**57.9**	47.5	52.7
EN + SP	44.9	60.5	**53.9**
EN + AR + SP	55.3	**61.7**	52.6

As expected, the models trained on a single language perform the best on the training data's language. For instance, the model trained in English performs the best for the English test data. This is due to the semantic differences and the implicit biases in each dataset. The results clearly indicate that training with data from different languages significantly

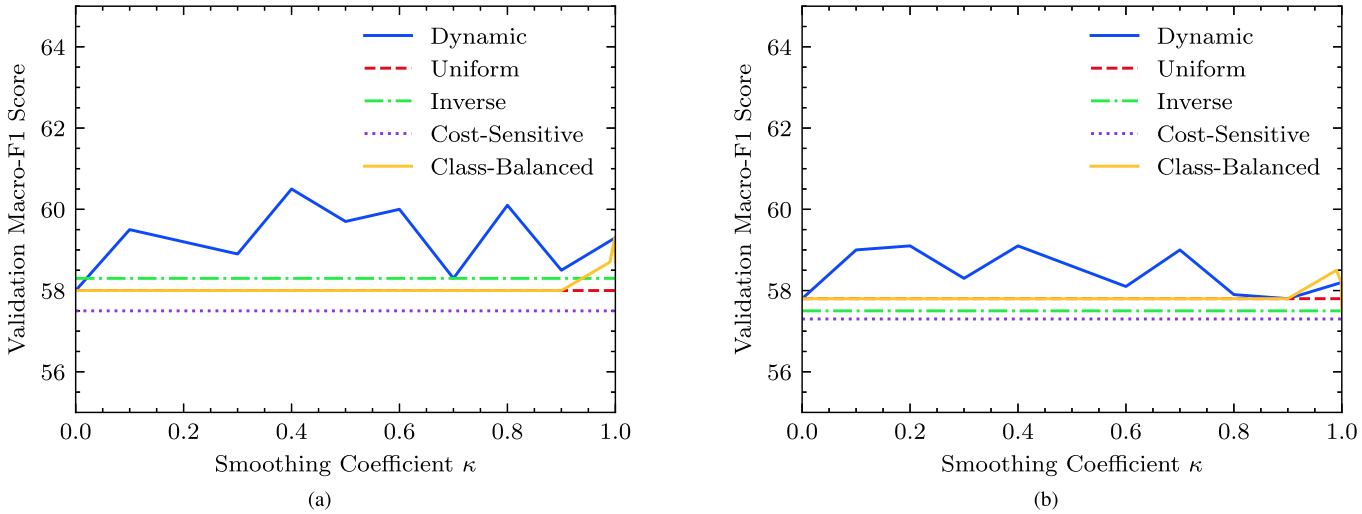

Fig. 3. Comparison of the weighting methods for imbalanced classification on the data from the SemEval emotion classification competition [1]. Figure is best viewed in color. (a) Only English data. (b) Combined Arabic, English, and Spanish data.

improves the performance of our model. For English test data, including Arabic to English training data improves the model more than including Spanish. For Arabic test data, including Spanish to Arabic training data improves the model more than including English. For Arabic test data, using Arabic and Spanish training data combined performs the best. For English test data, using data from all the three languages performs the best. For Spanish, including English data to Spanish training data improves the model more than including Arabic. For Spanish test data, using English and Spanish training data combined performs the best and including Arabic to these data lowers the performance.

Our cross-lingual experimental results are consistent for the models that are trained on single-language datasets with the semantic similarity atlas of the languages [48]. For example, English and Spanish are significantly more similar to each other than to the Arabic language. Among the models that are trained on a single language, English and Spanish training datasets score the best for each other's test data compared with Arabic. For English test data, Spanish training data score 7.0% (absolute) macro-f1 more than the Arabic. For Spanish test data, the model trained on English training data scores 5.5% (absolute) macro-f1 more than Arabic. For Arabic, which is closer to English than Spanish in the similarity atlas [48], training with English data results in 2.7% (absolute) macro-f1 gain compared with training with Spanish. Note that the models perform promising even for unseen languages, for example, the model that is trained on English and Spanish data and tested on the unseen Arabic validation data perform with 3% to 11% (absolute) less macro-f1 score compared with the baselines and our best model trained on all the three languages. The model tested on English data, which is trained on the rest of the languages, performed 2% (absolute) macro-f1 better than the Tw-StAR baseline and 10.9% (absolute) macro-f1 worse compared with the best English model trained on all three languages. For the model tested on the Spanish validation data, which is trained on the rest of the languages, the model performs 0.4% (absolute) macro-f1 better than the Tw-StAR baseline and 13.4% (absolute) macro-f1 worse than our best model that is trained on all the three languages. Note that these cross-lingual scores are obtained on the unseen validation sets of the datasets to prevent the test leak, unlike the baselines, where they are tested on the test set.

E. Influence of Dynamic Weighting

Here, we analyze the hyperparameter selection of our dynamic weighting method and compare it with the existing weighting methods that are proposed to remedy the class imbalance.

Fig. 3 illustrates the comparison of different weighting methods in the literature and our dynamic weighting method. We use the parameters of the best model except for κ, which is the smoothing hyperparameter for dynamic weighting. For dynamic weighting method, we experiment with different $\kappa \in [0, 1]$ with 0.1 spacing. For class-balanced focal loss term, we additionally experimented with the $\beta \in \{0.99, 0.999, 0.9999\}$ values as in [25]. Note that β is defined for $[0, 1)$, and thus, we did not experiment for $\beta = 1$. We show the β term of the class-balanced focal loss via the x-axis of Fig. 3, too, which controls the growth rate of the weight with respect to the number of instances belonging to each class. We experiment with uniform weighting that assigns equal importance to the losses from each class, that is, $\alpha_{t,a} = (1/w), \forall t \in \{1, 2, \ldots\}, a \in \{1, 2, \ldots w\}$. We also compare with the inverse loss, which is the inverse of the number of instances belonging to each class. Finally, we compare with the cost-sensitive loss [26].

Our dynamic weighting method demonstrates significant performance improvement, that is, $\approx$2.5% (absolute) macro-f1 improvement compared with uniform weighting and more improvements compared with the other methods on only English data. The only exception is the class-balanced weighting, for which our method achieves $\approx$1.2% (absolute) macro-f1 improvement compared with the best of the class-balanced weighting when $\beta = 0.999$. On the combined data,

TABLE III

ABLATION STUDY OF DIFFERENT NETWORK ARCHITECTURES AND LOSSES ON THE VALIDATION SET OF THE COMBINED ARABIC, ENGLISH, AND SPANISH DATA. RL AND AP INDICATE LABEL RANKING LOSS [52] AND AVERAGE PRECISION, RESPECTIVELY. "ZERO-ONE" AND "COVERAGE" INDICATE ZERO-ONE ERROR AND COVERAGE ERROR, RESPECTIVELY. LOWER IS BETTER FOR ZERO-ONE ERROR, COVERAGE ERROR, AND RANKING LOSS, AND HIGHER IS BETTER FOR THE REST

	Precision		Recall		F1		Ranking Based			
Method	Macro	Micro	Macro	Micro	Macro	Micro	Zero-One	Coverage	RL	AP
Ours	**56.4**	62.0	64.1	**73.5**	**59.4**	**67.2**	**0.7787**	**3.51**	**0.0994**	**62.7**
Ours w/o FL	54.2	**62.4**	60.6	69.7	57.1	65.9	0.7814	3.54	0.1021	61.1
Ours w/o DW	53.7	60.4	64.3	71.4	57.8	65.4	0.7814	3.52	0.0998	62.4
Ours w/o FL+DW	52.2	59.7	**64.8**	72.7	57.4	65.6	0.8020	3.56	0.1039	60.7
Uni-LSTM	54.1	60.6	61.7	71.0	57.1	65.4	0.7967	3.53	0.1009	60.4
Bi-RNN	52.4	59.9	61.8	70.9	56.4	65.0	0.8086	3.62	0.1069	59.1
Bi-GRU	54.5	61.3	62.6	72.6	57.7	66.5	0.7854	3.55	0.1038	61.9
XML-CNN [51]	40.8	44.6	62.9	70.2	47.8	54.5	0.9176	4.18	0.1578	48.0

the dynamic weighting achieves 1.3% macro-f1 improvement when $\kappa = 0.4$ compared with the uniform weighting and more improvements compared with the other methods. The only exception is the class-balanced weighting, for which our method achieves 0.6% (absolute) macro-f1 improvement when $\beta = 0.99$.

Although there exist fluctuations with respect to κ hyperparameter, it performs no worse than the default uniform weighting for any of the κ values for both only English and combined Arabic, English, and Spanish data. Note that when $\kappa = 0$, the dynamic weighting method is equivalent to uniform weighting since the ϕ parameter is never updated. Our method achieves its best value at $\kappa = 0.4$ for both the datasets.

F. Ablation Study

Here, we perform an ablation study to assess the performance gains obtained by the components in our method. We experiment with our best model, including both focal loss and dynamic weighting (Ours), our model with dynamic weighting only (Ours w/o FL), our model with focal loss only (Ours w/o DW), bidirectional recurrent neural networks (Bi-RNNs) [37], bidirectional gated recurrent unit (Bi-GRU) [49], unidirectional LSTM (Uni-LSTM) [38], bidirectional LSTM (Bi-LSTM) [50], and XML-CNN [51] model that is proposed for the extreme multi-label classification tasks with more than thousand labels. Note that all the models include focal loss and dynamic weighting unless otherwise stated.

Table III presents the results on the validation set of the combined Arabic, English, and Spanish data when the recurrent component is changed with other models and the loss changed with the standard cross-entropy loss. For each row, we only change the loss or the model. We keep all other hyperparameters as is. Our model with focal loss and dynamic weighting performs significantly better compared with others and outperforms them in eight of ten metrics. Adding focal loss improves the model by 2.2% (absolute) macro-f1, and adding dynamic weighting improves the model by 2.7% (absolute) macro-f1. To further understand the effects of focal loss and dynamic weighting on all classes, we first conduct Friedman test by calculating per class F1 scores for each classifier on Table III. With $p < 0.01$, we reject the null hypothesis that all models perform the same. Following the highly significant Friedman test, we perform the Wilcoxon signed-rank test to out model using the same per-class F1 setup. The results show us that there is no statistical significance in using focal loss or dynamic weighting by themselves ($p > 0.05$), but combining both these methods causes a significant increase in performance ($p < 0.05$). Unidirectional LSTM, which runs on the sentences only in the forward direction, performs 2.3% (absolute) macro-f1 worse compared with its bidirectional variant. Although GRU works better than RNN, it performs 1.7% worse compared with the bidirectional LSTM. XML-CNN, which is a CNN-based model and performs significantly worse compared with the other models.

G. Running Time Comparisons

Table IV shows elapsed times per epoch and in total. We have performed measurements through NVIDIA GeForce GTX 1080 Ti. Our best model, containing all components, takes about 5 min to run. The methods we introduced, that is, dynamic weighting and focal loss for multi-label classification, incur negligible running time cost. The increase in time per epoch is no more than 0.7 s, and the increase in total time is no more than 5 s. Moreover, these methods do not affect the asymptotic time complexity for Big-O notation.

Furthermore, XML-CNN, Bi-RNN, and Uni-LSTM reduce the running time by 8 s at most per epoch compared with our best model. These models also reduce the total time by 1 min and 56 s at most compared with our best model. However, our best model is significantly more accurate than other models, as shown in Table III. Note that XLM-RoBERTa is considerably slower due to the number of trained parameters.

H. Individual Class Performances

In this section, we analyze the performance of our method for individual classes.

TABLE IV

RUNNING TIME MEASUREMENTS OF DIFFERENT MODELS. TIME PER EPOCH DEPICTS THE TIME IT TAKES FOR THE MODEL TO APPLY FORWARD AND BACKWARD OPERATIONS FOR EACH TRAINING DATA ONCE. TOTAL TIME IS THE TOTAL ELAPSED TIME OF EXECUTION STARTING FROM THE DATA READ OPERATION AND ENDING WITH THE CONVERGE OF THE MODEL, WHICH CAN HAPPEN WITH EARLY STOPPING

Method	Time Per Epoch	Total Time
Ours	25.3 secs	5 mins 4 secs
Ours w/o FL	24.6 secs	4 mins 59 secs
Ours w/o DW	25.2 secs	4 mins 58 secs
Ours w/o FL+DW	25.0 secs	5 mins 1 secs
Uni-LSTM	17.6 secs	3 mins 10 secs
Bi-RNN	18.4 secs	4 mins 7 secs
Bi-GRU	21.5 secs	3 mins 28 secs
XML-CNN [51]	17.1 secs	3 mins 30 secs
XLM-RoBERTa [23]	306.2 secs	81 mins 3 secs

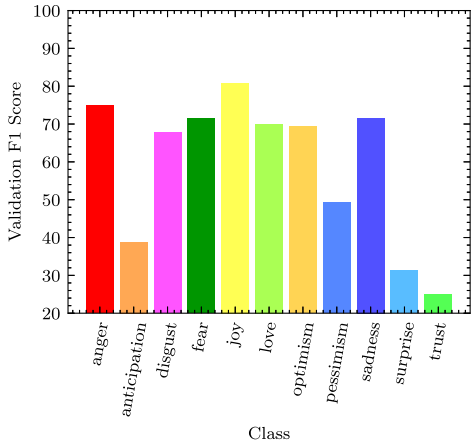

Fig. 4. Per-class F1 scores of the validation set of the combined Arabic, English, and Spanish data. Figure is best viewed in color.

Fig. 4 illustrates the validation F1 score for all classes on the combined data using the best model on combined data obtained in Section IV-C. The model performs the best for the *joy* class with 80.9% macro-f1 and performs the worst for the *trust* class with 25.0% macro-f1. The *surprise* and *trust* classes perform the worst among all as expected since their number of instances is the least. Discrimination of the *optimism* is significantly better than the *pessimism* as the number of instances in the *optimism* class is significantly higher than the number of instances in the *pessimism* class. Interestingly, the *anticipation* class is the third worst performing class, although it is not the third in terms of rarity, which is consistent with the results of the NVIDIA study [18]. Our model performs around 70% for the rest of the classes, that is, *anger*, *disgust*, *fear*, *joy*, *love*, *optimism*, and *sadness*.

V. CONCLUSION

We have investigated the cross-lingual sentiment analysis in multi-label setting. We have introduced a system that performs sentiment analysis in 100 different languages. To cope with the inherent class imbalance problem of multi-label classification, we have introduced a dynamic weighting method to remedy the inherent class imbalance problem of multi-label classification. This method balances the loss contribution of the classes as the training progresses, unlike the static weighting methods that assign non-changing weights to the classes. We have adapted the focal loss to the multi-label setting from the single-label object recognition literature. Moreover, we have derived a macro-f1 maximization method in linear time complexity for choosing class-specific thresholds to produce predictions. Our system has achieved the state-of-the-art performance in seven of nine metrics in three different languages on the SemEval emotion classification competition [1]. We have demonstrated the performance gains compared with the first ranking methods in the SemEval emotion classification competition [1] and the common baselines. We have also evaluated our method in the cross-lingual setting. We have demonstrated the performance gains obtained by the dynamic weighting and analyzed the effects of our method's components through an ablation study.

REFERENCES

[1] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 task 1: Affect in tweets," in *Proc. 12th Int. Workshop Semantic Eval. (Semeval)*, New Orleans, LA, USA, 2018, pp. 1–17.

[2] L. Zhu, W. Li, Y. Shi, and K. Guo, "SentiVec: Learning sentiment-context vector via kernel optimization function for sentiment analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2561–2572, Jun. 2021.

[3] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[4] D. Wang *et al.*, "Coarse alignment of topic and sentiment: A unified model for cross-lingual sentiment classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 736–747, Feb. 2021.

[5] E. Cambria, S. Poria, A. Hussain, and B. Liu, "Computational intelligence for affective computing and sentiment analysis [guest editorial]," *IEEE Comput. Intell. Mag.*, vol. 14, no. 2, pp. 16–17, May 2019.

[6] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of Emotion*. Amsterdam, The Netherlands: Elsevier, 1980.

[7] E. Cambria, A. Livingstone, and A. Hussain, "The hourglass of emotions," in *Cognitive Behavioural Systems*. Berlin, Germany: Springer, 2012, pp. 144–157.

[8] Y. Susanto, A. G. Livingstone, B. C. Ng, E. Cambria, and E. Cambria, "The hourglass model revisited," *IEEE Intell. Syst.*, vol. 35, no. 5, pp. 96–102, Sep. 2020.

[9] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2409–2429, Jul. 2020.

[10] T. Wei and Y.-F. Li, "Does tail label help for large-scale multi-label learning?" *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2315–2324, Jul. 2020.

[11] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[12] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, "Multilingual sentiment analysis: From formal to informal and scarce resource languages," *Artif. Intell. Rev.*, vol. 48, no. 4, pp. 499–527, Dec. 2017.

[13] E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, "SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 105–114.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[16] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[17] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2017, pp. 1–13.

[18] N. Kant, R. Puri, N. Yakovenko, and B. Catanzaro, "Practical text classification with large pre-trained language models," 2018, *arXiv:1812.01207*. [Online]. Available: http://arxiv.org/abs/1812.01207

[19] J. Suttles and N. Ide, "Distant supervision for emotion classification with discrete binary values," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Berlin, Germany: Springer, 2013, pp. 121–136.

[20] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Proc. Int. AAAI Conf. Web Social Media*, 2011, vol. 5, no. 1, pp. 538–541.

[21] A. Balahur and M. Turchi, "Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 56–75, Jan. 2014.

[22] D. Vilares, H. Peng, R. Satapathy, and E. Cambria, "BabelSenticNet: A commonsense reasoning framework for multilingual sentiment analysis," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1292–1298.

[23] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*. [Online]. Available: http://arxiv.org/abs/1911.02116

[24] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5375–5384.

[25] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9268–9277.

[26] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, "Learning from imbalanced data sets with weighted cross-entropy function," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1937–1949, Oct. 2019.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[29] G. Badaro *et al.*, "EMA at SemEval-2018 task 1: Emotion mining for arabic," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 236–244.

[30] C. Baziotis *et al.*, "NTUA-SLP at SemEval-2018 task 1: Predicting affective content in tweets with deep attentive RNNs and transfer learning," 2018, *arXiv:1804.06658*. [Online]. Available: http://arxiv.org/abs/1804.06658

[31] G. Gee and E. Wang, "PsyML at SemEval-2018 task 1: Transfer learning for sentiment and emotion analysis," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 369–376.

[32] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A set of arabic word embedding models for use in arabic NLP," *Procedia Comput. Sci.*, vol. 117, pp. 256–265, Nov. 2017.

[33] H. Mulki, C. B. Ali, H. Haddad, and I. Babaoglu, "Tw-StAR at SemEval-2018 task 1: Preprocessing impact on multi-label emotion classification," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 167–171.

[34] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 1–45, Aug. 2011.

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[36] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," 2018, *arXiv:1808.06226*. [Online]. Available: http://arxiv.org/abs/1808.06226

[37] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[40] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*. [Online]. Available: http://arxiv.org/abs/1607.04606

[41] A. E. Samy, S. R. El-Beltagy, and E. Hassanien, "A context integrated model for multi-label emotion detection," *Procedia Comput. Sci.*, vol. 142, pp. 61–71, Jan. 2018.

[42] N. Alswaidan and M. E. B. Menai, "Hybrid feature model for emotion recognition in arabic text," *IEEE Access*, vol. 8, pp. 37843–37854, 2020.

[43] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: https://arxiv.org/abs/1907.11692

[44] J.-A. González, L.-F. Hurtado, and F. Pla, "ELiRF-UPV at IroSvA: Transformer encoders for Spanish irony detection," in *Proc. IberLEF@SEPLN*, 2019, pp. 1–7.

[45] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2623–2631.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[48] L. K. Senel, I. Utlu, V. Yücesoy, A. Koc, and T. Cukur, "Generating semantic similarity atlas for natural languages," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 795–799.

[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[50] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[51] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 115–124.

[52] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Mining Multi-Label Data*. Boston, MA, USA: Springer, 2010, pp. 667–685, doi: 10.1007/978-0-387-09823-4_34.

Selim F. Yilmaz (Graduate Student Member, IEEE) received the B.S. degree (Hons.) in computer engineering from Bilkent University, Ankara, Turkey, in 2019, where he is currently pursuing the M.S. degree with the Department of Electrical and Electronics Engineering.

His current research interests include federated learning, anomaly detection, and natural language processing.

E. Batuhan Kaynak received the B.S. degree (Hons.) in computer engineering from Bilkent University, Ankara, Turkey, in 2019, where he is currently pursuing the M.S. degree with the Department of Computer Engineering.

His current research interests include affective computing, computer vision, and natural language processing.

Aykut Koç (Senior Member, IEEE) received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2005, and the M.S. degree in electrical engineering, the M.S. degree in management science and engineering, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2007, 2009, and 2011, respectively.

He is a Faculty Member with the Electrical and Electronics Engineering Department and with the National Magnetic Resonance Research Center (UMRAM), Bilkent University. Before joining Bilkent University in 2019, he worked in the founding team of researchers who raised the ASELSAN Research Center from ground-up. He founded and managed one of the departments of the ASELSAN Research Center with principal investigator and research manager capacities. He also taught part-time with the Department of Electrical and Electronics Engineering, Middle East Technical University (METU), Ankara. He has authored or coauthored more than 50 research articles, one book chapter, and four issued patents. His current research interests are in natural language processing and signal/image processing.

Hamdi Dibeklioğlu (Member, IEEE) received the Ph.D. degree from the University of Amsterdam, Amsterdam, The Netherlands, in 2014.

He is currently an Assistant Professor with the Computer Engineering Department, Bilkent University, Ankara, Turkey, and a Research Affiliate with the Pattern Recognition and Bioinformatics Group, Delft University of Technology, Delft, The Netherlands. Before joining Bilkent University, he was a Post-Doctoral Researcher with the Delft University of Technology. His research focuses on computer vision, pattern recognition, affective computing, and computer analysis of human behavior.

Dr. Dibeklioğlu is a Program Committee Member of several top-tier conferences in these areas. He was the Co-Chair of the Netherlands Conference on Computer Vision in 2015, the Local Arrangements Co-Chair of the European Conference on Computer Vision in 2016, the Publication Co-Chair of the European Conference on Computer Vision in 2018 and 2020, the Co-Chair of the eNTERFACE Workshop on Multimodal Interfaces in 2019, and the Area Chair of the IEEE International Conference on Automatic Face and Gesture Recognition in 2020 and the ACM International Conference on Multimodal Interaction in 2021.

Suleyman Serdar Kozat (Senior Member, IEEE) received the B.S. degree (Hons.) from Bilkent University, Ankara, Turkey, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2001 and 2004, respectively.

He joined the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, as a Research Staff Member and later became a Project Leader with the Pervasive Speech Technologies Group, where he focused on problems related to statistical signal processing and machine learning. He was a Research Associate with the Cryptography and Anti-Piracy Group, Microsoft Research, Redmond, WA, USA. He is currently a Professor with the Electrical and Electronics Engineering Department, Bilkent University. He has coauthored over 120 articles in refereed high-impact journals and conference proceedings and holds several patent inventions (currently used in several different Microsoft and IBM products, such as MSN and ViaVoice). He holds several patent inventions due to his research accomplishments with the IBM Thomas J. Watson Research Center and Microsoft Research. His current research interests include cybersecurity, anomaly detection, big data, data intelligence, adaptive filtering, and machine learning algorithms for signal processing.

Dr. Kozat received many international and national awards. He is the Elected President of the IEEE Signal Processing Society, Turkey Chapter.