# Text Emotion Distribution Learning from Small Sample: A Meta-Learning Approach

**Zhenjie Zhao** **Xiaojuan Ma**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{zzhaoao, mxj}@cse.ust.hk

## Abstract

Text emotion distribution learning (EDL) aims to develop models that can predict the intensity values of a sentence across a set of emotion categories. Existing methods based on supervised learning require a large amount of well-labelled training data, which is difficult to obtain due to inconsistent perception of fine-grained emotion intensity. In this paper, we propose a meta-learning approach to learn text emotion distributions from a small sample. Specifically, we propose to learn low-rank sentence embeddings by tensor decomposition to capture their contextual semantic similarity, and use K-nearest neighbors (KNNs) of each sentence in the embedding space to generate sample clusters. We then train a meta-learner that can adapt to new data with only a few training samples on the clusters, and further fit the meta-learner on KNNs of a testing sample for EDL. In this way, we effectively augment the learning ability of a model on the small sample. To demonstrate the performance, we compare the proposed approach with state-of-the-art EDL methods on a widely used EDL dataset: SemEval 2007 Task 14 (Strapparava and Mihalcea, 2007). Results show the superiority of our method on small-sample emotion distribution learning.

## 1 Introduction

Analyzing emotions in text automatically is an important topic (Yadollahi et al., 2017) with widely used applications such as classifying e-commerce products reviews (Rao et al., 2016) and developing emotionally intelligent chatbots for healthcare (Fadhil and Gabrielli, 2017), to name a few. Text emotion analysis aims to recognize writers' emotional states towards particular topics or subjects (Yadollahi et al., 2017), and has attracted considerable research efforts in the last few decades (Canales and Martínez-Barco, 2014; Abdul-Mageed and Ungar, 2017; Yu et al., 2018; Zhang et al., 2018).

Existing works cast text emotion analysis into three types of tasks: single label learning (SLL), multi-label learning (MLL), and label distribution learning (LDL). In SLL, one particular name in an emotion category (referred as label in the following) is predicted (Abdul-Mageed and Ungar, 2017), e.g., joyful, angry, or sad. However, due to the correlation among different emotions, one sentence can potentially contain multiple different emotions (Yu et al., 2018). Therefore, a more practical way is to assign multiple labels to a sentence (Yu et al., 2018), namely, MLL. Label distribution learning, which is called emotion distribution learning (EDL) in emotion mining (Zhou et al., 2016), goes a step further and assigns one intensity value to each emotion, which is necessary to encode fine-grained information for comparing strength of different emotions, etc.

In general, there are three approaches to text emotion analysis tasks: lexicon-based (Staiano and Guerini, 2014), learning-based (Zhang et al., 2018), and the combination of both methods (Mudinas et al., 2012). In the lexicon-based case, one needs to collect an emotion lexicon corpus, e.g., WordNet-Affect (Strapparava and Valitutti, 2004), SentiWordNet (Esuli and Sebastiani, 2006), and applies different counting methods to aggregate the occurrences of words associated with various emotions (Staiano and Guerini, 2014). Learning-based methods, in contrast, mostly frame emotion analysis as a supervised learning problem, which requires annotated text data with emotion labels and extraction of proper sentence features. Learning-based approaches alone, or combined with lexicon-based methods, usually produce state-of-the-art (SOTA) performance (Mudinas et al., 2012; Zhang et al., 2018), and thus are widely used nowadays.

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3957–3967,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

However, learning-based methods usually demand a large amount of annotated data to train models, which has become one of the performance bottlenecks. EDL in particular aims to decode fine-grained composition and magnitude of emotions in text, the human perception of which can be highly subtle and personal (Volkova et al., 2010). It is difficult, if not impossible, to collect a large-scale emotion distribution dataset with consistent, clean human labels. Therefore, developing techniques to learn from a small sample is critical for the practicality of emotion distribution analysis.

In this paper, based on meta-learning (Vilalta and Drissi, 2002; Finn et al., 2017), we propose an efficient approach to learn text emotion distribution from a small sample. To make the most of a small labeled dataset, we propose to use the K-nearest semantically similar neighbors (KNNs) of each training sample to cluster the training data, and train a meta-learner that can adapt to new testing data with only a few samples on the clusters. We can then fit the meta-learner on KNNs of each testing sample. Learning semantic similarity of sentences usually requires a large amount of data (Le and Mikolov, 2014). We propose to learn low-rank embeddings of sentences by tensor decomposition to capture their contextual semantic similarity, which works well regardless of the size of documents (Hosseinimotlagh and Papalexakis, 2018). We evaluate the proposed approach on a widely used text emotion distribution dataset: SemEval 2007 Task 14 (Strapparava and Mihalcea, 2007), and show that it outperforms the existing SOTA methods for small sample EDLs.

The contributions of this paper are: 1) we propose a novel meta-learning framework to learn text emotion distribution from a small sample; 2) we propose to learn low-rank embeddings of sentences by tensor decomposition to find semantically similar neighbors for training and adapting a meta-learner.

## 2 Related Works

We briefly review three related areas that motivate this work, including emotion distribution learning, text representation learning, and learning from a small sample.

### 2.1 Emotion Distribution Learning

Emotion distribution learning of text (Zhou et al., 2016; Zhang et al., 2018) is a recently proposed task that tries to predict the intensity values of a sentence across a set of emotion categories. Such information is important for understanding the fine-grained emotion information (Zhou et al., 2016; Zhang et al., 2018). For example, one sentence can usually invoke multiple emotional states with different levels, and existing approaches of SLL and MLL are inadequate for capturing multi-label, multi-intensity information. LDL is more suitable for such scenarios. In general, LDL methods can be classified into three categories: problem transformation, algorithm adaption, and specialized algorithms (Geng and Ji, 2013). PT-Bayes and PT-SVM are typical problem transformation methods that transform the LDL problem into an SLL problem, and use Bayes classifier and SVM to predict label distribution, respectively. AA-KNN and AA-BP are algorithm adaption methods that extend K-nearest neighbors (KNNs) and back-propagation (BP) neural networks. SA-LDSVR, SA-IIS, SA-BFGS, SA-CPNN are specialized LDL algorithms that directly parametrize and optimize LDL objectives (Geng and Ji, 2013). In terms of EDL, two different models have been proposed recently, including a maximum entropy model (Zhou et al., 2016) and a convolutional neural network model (Zhang et al., 2018), and the latter one is the SOTA.

### 2.2 Text Representation Learning

Learning numerical representation of text is usually the first step for learning-based emotion analysis, which can be categorized at two levels: word/phrase and sentence/document. At the word/phrase level, for example, Mikolov et al. (2013) propose word2vec which uses a three-layer neural network, i.e., the input layer, the projection layer, and the output layer, to learn word vectors from a large corpora in an unsupervised manner. The projection layer maps one-hot encoded input to a low-dimension vector, which is used to predict a word in the output layer. The learned word vectors of the projection layer are shown to be an effective word representation. At the sentence/document level, similar to word2vec, doc2vec (Le and Mikolov, 2014) treats sentences and context words in the same way in the input layer, and uses a three-layer neural network to learn the vector representation of sentences. Similarly, the skip-thought vector method (Kiros et al., 2015) uses sentences directly for prediction tasks, and ignores context words used in doc2vec. Recent works show that using attention mechanism, transformer in particular (Vaswani et al., 2017), is an effective way to learn universal sentence representation with unlabeled text, such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019). There are other ways to improve the discriminative ability of the learned features through supervised learning, such as task transfer methods like InferSent (Conneau et al., 2017).

### 2.3 Learning from A Small Sample

The success of machine learning models, especially deep learning models, heavily depends on a large amount of labeled data. But acquiring training data is usually difficult and expensive (Johnson et al., 2018). Therefore, considerable research efforts on small sample learning are emerging recently (Lake et al., 2015; Shu et al., 2018). There are generally two approaches for learning from a small sample (Shu et al., 2018): concept learning and experience learning. Concept learning means analyzing the structure of example samples by imitating human abilities like imagination, synthesis, and analysis (Lake et al., 2015; Shu et al., 2018). Experience learning means augmenting or transferring learning experience from other data or models, including data augmentation, model fine-tuning, model compression, and meta-learning (Shu et al., 2018). Meta-learning in particular is an effective way of augmenting the experience learning ability (Thrun and Pratt, 1998; Vilalta and Drissi, 2002; Finn et al., 2017).

#### 2.3.1 Meta-Learning

Meta-learning aims to increase the learning performance through experience sharing among tasks (Thrun and Pratt, 1998; Vilalta and Drissi, 2002; Finn et al., 2017). One kind of meta-learning is to train a meta-learner to update the parameters or rules of the learning models (Bengio et al., 1991). However, the complexity of this approach is usually high (Bengio et al., 1991; Finn et al., 2017). Another approach is to train a memory neural network, such as recurrent neural networks (Santoro et al., 2016), to keep a record of the different experiences among different tasks. In particular, Finn et al. (2017) find that it is possible to optimize a learner to adapt new tasks quickly only by gradient descent. In this approach, no additional parameters are needed. Recently, researchers have also extended this method to unsupervised learning (Hsu et al., 2019) and online learning scenarios (Finn et al., 2019). Meta-learning has been used for few-shot image recognition (Lee et al., 2019), agent navigation (Finn et al., 2017), low-resource machine translation (Gu et al., 2018), query generation (Huang et al., 2018), and so on.

Existing EDL methods are designed with traditional machine learning paradigms (Geng and Ji, 2013; Zhou et al., 2016; Zhang et al., 2018), which may not work well on a small sample. Due to the difficulty of annotating emotion distribution data, developing techniques of small sample EDLs is thus critical for text emotion analysis. Transferring experience with pre-trained models (Shu et al., 2018) can be an effective way to boost natural language tasks. But such methods may need an additional large training corpora, and fine-tuning models on specific domains is non-trivial (Le and Mikolov, 2014). In this paper, we are interested in how to only use the labeled data for small sample EDLs (Shu et al., 2018).

## 3 Problem Formulation

### 3.1 Emotion Distribution Learning (EDL)

We denote one sample as $(s, \mathbf{y})$, where $s$ is a sentence or document, $\mathbf{y} = (y^1, y^2, \ldots, y^C)$ is its emotion distribution label, $y^j$ denotes the intensity value of the $j$-th emotion, $C$ is the number of discrete emotions, and $\sum_j^C y^j = 1$. For EDL, we expect to train a model $f : \mathcal{S} \mapsto \mathcal{Y}$ that can accurately map any sentence $s$ in semantic space $\mathcal{S}$ to its entailed emotion distribution $\mathbf{y}$ in emotion distribution space $\mathcal{Y}$.

### 3.2 EDL from A Small Sample

Given a set of training samples $\mathcal{D}_{\text{train}} = \{(s_i, \mathbf{y}_i)\}_{i=1}^M$, we are interested in how to train a model $f$ that can predict emotion distribution effectively on a set of testing samples $\mathcal{D}_{\text{test}} = \{(s_{\text{i}}, \mathbf{y}_{\text{i}})\}_{i=1}^N$, where $M$ is the number of training samples and is a small number (e.g., 50), $N$ is the number of testing samples.

For EDL from a small sample, we adopt the experience learning approach (Shu et al., 2018) by considering: 1) how to augment learning ability on a small number of training samples, and 2) how to learn semantic similarity on $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ so that we can use K-nearest neighbors in $\mathcal{D}_{\text{train}}$ to predict the emotion distribution of a testing sample in $\mathcal{D}_{\text{test}}$.

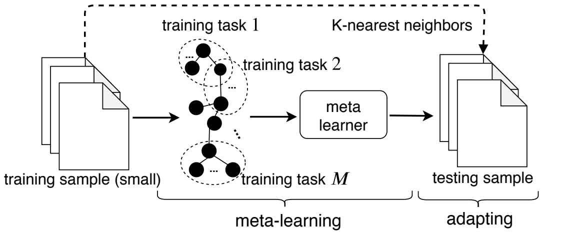

Figure 1: Illustration diagram of the proposed EDL method.

## 4 Method

As shown in Figure 1, to augment the learning ability on a small sample, we first partition the training data by finding K-nearest neighbors (KNNs) of each sample based on semantic similarity, and treat each $K$ samples as a cluster. We then train a meta-learner that is optimized to adapt to new data with only a few training samples on the clusters. During testing, we fit the meta-learner with KNNs of a testing sample, where KNNs are found in the training data.

### 4.1 Meta-Learning Preliminary

A meta-learning algorithm consists of a set of training tasks $\mathcal{T}_{\text{train}} = \{T_i\}_{i=1}^{M}$, a set of testing tasks $\mathcal{T}_{\text{test}} = \{T_i\}_{i=1}^{N}$, and a model $f_\theta$ [1], where $M$ and $N$ are the numbers of the training and testing tasks separately, and $\theta$ is the parameters of $f$. Each task $T$ consists of a set of training samples called supporting set $S$ and a set of testing samples called querying set $Q$, namely, $T = \{S, Q\}$. The goal of meta-learning is to train a learner $f_\theta$ using $\mathcal{T}_{\text{train}}$ so that given a new testing task $T$ in $\mathcal{T}_{\text{test}}$, it can perform well on the query set $Q$ of $T$ by fine-tuning it with only a few samples in the support set $S$.

### 4.2 EDL via Meta-Learning

We propose a meta-learning framework for EDL, including task generation from a small sample, meta-learner training, and meta-learner adaption for predicting testing samples.

#### 4.2.1 Task Generation

For each sample in $\mathcal{D}_{\text{train}}$, we first find its KNNs by semantic similarity, and treat the $K$ samples as a training task. Therefore, samples in each task can be overlapped, and we use this way to increase the number of tasks. For each task, we randomly select $K/2$ samples as the support set, and use the remaining ones as the query set, i.e., the sizes of $S$ and $Q$ are both $K/2$. Therefore, for $M$ samples in $\mathcal{D}_{\text{train}}$, we can generate $M$ training tasks $\mathcal{T}_{\text{train}}$. The intuition is that semantically similar sentences are more likely to have similar emotion distribution patterns.

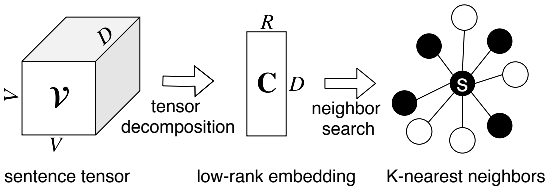

Figure 2: Tensor decomposition to find KNNs of each sentence.

**Algorithm 1:** Text EDL via Meta-Learning

```
Data: training sample $\mathcal{D}_{\text{train}}$ and testing
      sample $\mathcal{D}_{\text{test}}$
Result: learner $f_{\theta_s}$ for each sentence $s$ in
        $\mathcal{D}_{\text{test}}$
initialize hyper-parameters
  $\alpha, \beta, \gamma, H, R, K, niter, L$;
// Task Generation
calculate sentence embeddings $\mathbf{C}$ with
  formula (1), and for each sample in $\mathcal{D}_{\text{train}}$,
  find its KNNs to get $\mathcal{T}_{\text{train}}$;
// Meta-Learner Training
initialize the meta-learner $f$ with parameter $\theta$;
for $iter$ in $1, 2, \ldots, niter$ do
    sample $L$ tasks from $\mathcal{T}_{train}$ randomly;
    for $l$ in $1, 2, \ldots, L$ do
        $\theta'_l \leftarrow \theta - \alpha \, \nabla_\theta \, \mathcal{L}_{T_l(S)}(f_\theta)$
    end
    $\theta \leftarrow \theta - \beta \, \nabla_\theta \sum_l \mathcal{L}_{T_l(Q)}(f_{\theta'_l})$;
end
// Adapting Meta-Learner
for each sentence $s$ in $\mathcal{D}_{test}$ do
    find KNNs $\mathcal{K}_s$ of $s$ in $\mathcal{D}_{\text{train}}$;
    $\theta_s \leftarrow \theta - \nabla_\theta \mathcal{L}_{\mathcal{K}_s}(f_\theta)$
end
```

**Low-rank embedding by tensor decomposition** Contextual patterns of words can be used to measure semantic similarity for emotion analysis (Staiano and Guerini, 2014; Mikolov et al., 2013). Traditional embedding approaches, e.g., doc2vec, usually require a large amount of data (Le and

[1] To be consistent with the existing literature, we refer to the *model* as the *learner* in the rest of the paper.

Mikolov, 2014). We propose to use low-rank embeddings of sentences mined by tensor decomposition, which can obtain text embeddings regardless of the corpus size (Hosseinimotlagh and Papalexakis, 2018). As shown in Figure 2, considering $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$, we first build a vocabulary for it, namely, $w_1, w_2, \dots, w_V$, where $V$ is the number of words. For each sentence $s$ in $\mathcal{D}$, we count the word-word co-occurrence in a small window $H$, and build a binary matrix $\mathbf{V}_s \in [0,1]^{V \times V}$. In particular, $\mathbf{V}_s(i,j) = 1$ indicates word $w_i$ and $w_j$ co-occurs in $s$ within a small window $H$ at least once. In this way, we can capture the semantic patterns of a sentence in $\mathbf{V}_s$. In addition, this approach can also deal with negation issue to some extent, which is important for sentiment analysis (Reitan et al., 2015). Because negation sentences usually have negation words, they will have a different word-word co-occurrence pattern with that of normal sentences, which will result in different embeddings. Therefore, the tensor decomposition method may put sentences with and without negation words into different clusters. Afterwards, we stack all $\mathbf{V}_s$ as a three-dimensional tensor $\boldsymbol{\mathcal{V}} \in [0,1]^{V \times V \times D}$, where $D = M + N$. We adopt the CANDECOMP/PARAFAC tensor decomposition method (Sidiropoulos et al., 2017) to find an approximation $\hat{\boldsymbol{\mathcal{V}}}$ of $\boldsymbol{\mathcal{V}}$:

$$\hat{\boldsymbol{\mathcal{V}}} = \sum_{r=1}^{R} \mathbf{w}_r \otimes \mathbf{w}_r \otimes \mathbf{s}_r, \tag{1}$$

such that the Frobenius norm $||\hat{\boldsymbol{\mathcal{V}}} - \boldsymbol{\mathcal{V}}||_F$ is minimal, where $\mathbf{w}_r \in \mathbb{R}^V$, $\mathbf{s}_r \in \mathbb{R}^D$, $R$ is the rank, and $\otimes$ is the outer product, namely, $\mathbf{w}_r \otimes \mathbf{w}_r \otimes \mathbf{s}_r$ being a three-dimensional tensor, and $\mathbf{w}_r \otimes \mathbf{w}_r \otimes \mathbf{s}_r(i,j,k) = \mathbf{w}_r(i) \cdot \mathbf{w}_r(j) \cdot \mathbf{s}_r(k)$. We use both training and testing datasets for embedding, which follows the general practice of previous literature (Zhou et al., 2016; Zhang et al., 2018). In such a case, it is possible to infer a complete vocabulary of the corpus by making use of both the training and testing dataset. If some testing data are not available when the tensor is built, there exist tensor decomposition methods for dealing with streaming data scenarios (Gujral et al., 2018), namely, the testing data can be updated continuously over time. Our framework can easily utilize such advanced methods to generalize to sentences with out-of-vocabulary words.

With the tensor decomposition, we can find low-rank embeddings of sentences that capture the similarity of contextual patterns (Hosseinimotlagh and Papalexakis, 2018). In particular, $\mathbf{C} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_R] \in \mathbb{R}^{D \times R}$, where the $s$-row of $\mathbf{C} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \dots, \mathbf{c}_D^T]^T$, $\mathbf{c}_s^T$, is the embedding vector of the sentence $s$. We measure the similarity of two sentences $i, j$ by Euclidean distance: $d(i,j) = \sqrt{\sum_{r=1}^{R} (\mathbf{c}_i(r) - \mathbf{c}_j(r))^2}$, where $\mathbf{c}(r)$ denotes the $r$-th element of $\mathbf{c}$.

#### 4.2.2 Meta-Learner Training

We can train a meta-learner on the generated $\mathcal{T}_{\text{train}}$.

**Learner.** We use a Convolutional Neural Network (CNN) model as the basic learner, which has shown good performance on text classification (Kim, 2014) and text emotion distribution learning (Zhang et al., 2018). The CNN learner has the same architecture with (Zhang et al., 2018). In particular, given the input sentence $s$, we first stack a matrix $\mathbf{X}$ with the word vector of each word (Mikolov et al., 2013). Then three convolution layers with kernel sizes $3, 4, 5$ are performed on $\mathbf{X}$ separately, and we then concatenate them together. A fully connected layer is then used, following with a soft-max operation to get the final emotion probability prediction. We use Kullback-Leibler (K-L) divergence $\mathcal{L}_{\text{K-L}}$ as the objective to measure the distance of predicted distribution $\hat{\mathbf{y}}$ and the true distribution $\hat{\mathbf{y}}$: $\mathcal{L}_{\text{K-L}} = \sum_{i=1}^{K} \frac{1}{K} \sum_{j=1}^{C} y_i^j \log \frac{y_i^j}{\hat{y}_i^j}$, where $K$ is the number of training samples. Similarly to (Zhang et al., 2018), we also optimize the classification accuracy of the dominant emotion by a cross-entropy (CE) objective, which is shown to improve EDL performance. In particular, $\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{K} \frac{1}{K} \sum_{j=1}^{C} \mathbb{1}(y_i^j) \log \hat{y}_i^j$, where $\mathbb{1}(y^j) = 1$ if $y^j$ is the maximal value of $\mathbf{y}$, otherwise $\mathbb{1}(y^j) = 0$. The final objective is a weighted sum of $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{CE}}$: $\mathcal{L} = \gamma \mathcal{L}_{\text{K-L}} + (1 - \gamma) \mathcal{L}_{\text{CE}}$, where $\gamma$ is the weight factor, $0 \leq \gamma \leq 1$.

**Training meta-learner.** To train the learner $f_\theta$, we optimize the parameters $\theta$ of the learner $f$ such that a small number of gradient steps on a new task will produce maximally effective behavior on that task, namely,

$$\min_{\theta} \sum_{T_i \sim \mathcal{T}_{\text{train}}} \mathcal{L}_{T_i(Q)}(f_{\theta_i'}), \tag{2}$$

where $T_i$ is a randomly sampled task in $\mathcal{T}_{\text{train}}$, $\theta_i'$ is the trained optimal parameters on task $T_i$. $T_i(Q)$ denotes that the loss is computed on the query set

$Q$. Equation (3) is optimized via stochastic gradient descent (SGD),

$$\theta \leftarrow \theta - \beta(\triangledown_\theta \sum_{T_i \sim \mathcal{T}_{\text{train}}} \mathcal{L}_{T_i(Q)}(f_{\theta'_i})), \tag{3}$$

where $\beta$ is the meta-learning rate. Similarly, SGD is used to compute $\theta'_i$, $\theta'_i \leftarrow \theta - \alpha \triangledown_\theta \mathcal{L}_{T_i(S)}(f_\theta)$, where $T_i(S)$ denotes that the loss is computed on $S$, and $\alpha$ is the learning rate.

### 4.2.3 Adapting Meta-Learner

The meta-learner $f_\theta$ is trained to adapt to new data with only a few semantically similar training samples. Therefore, given a testing sample $s$, we first find its K-nearest neighbors $\mathcal{K}_s$ in the training data $\mathcal{D}_{\text{train}}$, and then adapt $f_\theta$ on $\mathcal{K}_s$ by SGD:

$$\theta \leftarrow \theta - \alpha \triangledown_\theta \mathcal{L}_{\mathcal{K}_s}(f_\theta). \tag{4}$$

Here we use the learning rate $\alpha$ to be consistent with the meta-training procedure.

The overall algorithm is shown in Algorithm 1, where $\alpha$ is the learning rate, $\beta$ is the meta-learning rate, $\gamma$ is the weight value of the distribution and classification losses, $H$ is the window size for word-word co-occurrence counting, $R$ is the rank of tensor decomposition, $K$ is the number of nearest neighbors, $niter$ is the number of meta-training iterations, and $L$ is the number of tasks for each round of meta-training.

## 5 Experiment

To evaluate the proposed approach for learning emotion distribution from a small sample, we conduct intensive experiments on SemEval 2007 Task 14 (Strapparava and Mihalcea, 2007). To the best of our knowledge, this is the only publicly available English dataset with emotion distributions labeled by humans.

### 5.1 Experiment Setup

### 5.1.1 Dataset

SemEval 2007 Task 14 contains 1250 sentences of news headlines with 6 emotion intensities (anger, disgust, fear, joy, sadness, and surprise) labeled by humans. Each intensity value is a non-negative value. We normalize the annotated scores to get emotion distribution labels using the same procedure with Zhang et al. (2018). In particular, given an original intensity tuple $(l_1, l_2, \ldots, l_6)$, we first calculate the sum of all values, $l = \sum_{k=1}^{6} l_k$, and then normalize it to get a distribution as $(l_1/l, l_2/l, \ldots, l_6/l)$. If $l = 0$, then the distribution will be $(1/6, 1/6, \ldots, 1/6)$. It is possible to add another intensity value $l_7$ denoting the overall level of emotion to the original label; for example, if $l_7$ is 1, there is no emotion in the text, and if $l_7$ is 0, there is strong emotions. Similar to $l_1 \sim l_6$, $l_7$ can be annotated manually. We can then normalize $(l_1, l_2, \ldots, l_7)$ to get a new distribution. In this way, we can better model the cases of a sentence expressing very little emotion overall. For simplicity and fair comparison, we only use $l_1 \sim l_6$ in this paper.

### 5.1.2 Experiment Protocal

Because there is no publicly available small dataset for text emotion distribution evaluation, we follow existing practices (Hosseinimotlagh and Papalexakis, 2018) to simulate a small training set by randomly selecting $10\%$ of the samples and using the remaining ones for testing. We run 10-fold cross validation, and report the averaged results.

**Baselines.** We compare the proposed method (denoted as EDL-Meta) with several baseline methods, including the SOTA method EDL-CNN (Zhang et al., 2018), LDL methods, and document vectorization methods. EDL-CNN uses the basic learner of EDL-Meta, and can be seen as a special case of EDL-Meta when $K = 0$. Similar to Zhang et al. (2018), we also extract the penultimate layer of EDL-CNN as features, and fit several typical LDL methods: PT-Bayes, PT-SVM, AA-KNN, AA-BP, SA-LDSVR, SA-IIS, SA-BFGS, SA-CPNN for comparison. The basic principles of each method are briefly summarized in subsection 2.1. In addition, we apply three SOTA document vectorization methods – doc2vec (Le and Mikolov, 2014), InferSent (Conneau et al., 2017), and BERT (Devlin et al., 2019) – and use the extracted vectors to train a linear regressor to predict emotion distributions.

### 5.1.3 Evaluation Metrics

Evaluating the performance of distribution learning is challenging, because we need to measure the prediction results of fine-grained intensity values. In (Geng and Ji, 2013), the authors propose six metrics, and suggest that each metric may reflect certain aspects of an algorithm. A good algorithm should perform well on most of them. The six metrics contain Euclidean$^{\downarrow}$, Sørensen$^{\downarrow}$, Squared$\chi^{2\downarrow}$,

| Method | Euclidean$^{\downarrow}$ | Sørensen$^{\downarrow}$ | Squared$\chi^{2\downarrow}$ | K-L$^{\downarrow}$ | Fidelity$^{\uparrow}$ | Intersection$^{\uparrow}$ |
|---|---|---|---|---|---|---|
| PT-Bayes | 0.4870 | 0.4889 | 0.6707 | 0.7574 | 0.7330 | 0.5111 |
| PT-SVM | 0.5748 | 0.5706 | 0.8578 | 1.1327 | 0.6639 | 0.4294 |
| AA-KNN | 0.4248 | 0.4054 | 0.5181 | 0.8325 | 0.7986 | 0.5946 |
| AA-BP | 0.4620 | 0.4468 | 0.5945 | 0.7410 | 0.7662 | 0.5532 |
| SA-LDSVR | 0.4336 | 0.4238 | 0.5301 | 0.5828 | 0.7927 | 0.5762 |
| SA-IIS | 0.4402 | 0.4305 | 0.5456 | 0.5983 | 0.7842 | 0.5695 |
| SA-BFGS | 0.8403 | 0.6909 | 1.1563 | 7.7697 | 0.5056 | 0.3091 |
| SA-CPCNN | 0.4779 | 0.4600 | 0.6108 | 0.7365 | 0.7660 | 0.5400 |
| BERT | 0.4498 | 0.4378 | 0.5620 | 0.6207 | 0.7770 | 0.5622 |
| InferSent | 0.4630 | 0.4518 | 0.5882 | 0.6488 | 0.7662 | 0.5482 |
| doc2vec | 0.4458 | 0.4334 | 0.5552 | 0.6133 | 0.7799 | 0.5666 |
| EDL-CNN | 0.4450 | 0.4345 | 0.5553 | 0.6105 | 0.7794 | 0.5655 |
| EDL-Meta | **0.3949** | **0.3747** | **0.4478** | **0.5148** | **0.8319** | **0.6253** |

Table 1: Evaluation results on SemEval 2007 Task 14.

| | | |
|---|---|---|
| Distance | Euclidean$^{\downarrow}$ | $\sqrt{\sum_{i=1}^{C}(p_i - q_i)^2}$ |
| | Sørensen$^{\downarrow}$ | $\frac{\sum_{i=1}^{C}\lvert p_i - q_i\rvert}{\sum_{i=1}^{C}(p_i + q_i)}$ |
| | Squared$\chi^{2\downarrow}$ | $\sum_{i=1}^{C}\frac{(p_i - q_i)^2}{p_i + q_i}$ |
| | K-L$^{\downarrow}$ | $\sum_{i=1}^{C} p_i \log \frac{p_i}{q_i}$ |
| Similarly | Fidelity$^{\uparrow}$ | $\sum_{i=1}^{C}\sqrt{p_i q_i}$ |
| | Intersection$^{\uparrow}$ | $\sum_{i=1}^{C}\min(p_i, q_i)$ |

Table 2: Metrics for evaluating the performance of emotion distribution learning.

K-L$^{\downarrow}$, Fidelity$^{\uparrow}$, Intersection$^{\uparrow}$, and can be classified into two categories: distance metrics and similarity metrics (Geng and Ji, 2013). Distance metrics measure the distance between the predicted distribution and the true one, the smaller the better. Similarity metrics measure the similarity between the predicted distribution and the true one, the bigger the better. Here, $^{\uparrow}$ means the bigger the better, $^{\downarrow}$ means the smaller the better. The formulas for calculating all metrics are summarized in Table 2, where $(q_1, q_2, \ldots, q_C)$ is the predicted distribution, $(p_1, p_2, \ldots, p_C)$ is the true distribution, $C$ is the number of emotions.

### 5.1.4 Implementation

We use 300 dimension word vectors trained on Google News from (Mikolov et al., 2013) as initial input. Because SemEval 2007 Task 14 is mostly News headlines, the pre-trained word vectors are closer to it. We implement our algorithm by PyTorch [2]. Owing to EDL-CNN not being open-sourced, we also implement it by PyTorch. The implementation of doc2vec is adopted from gemsim [3]. BERT embedding (uncased base model, without fine-tuning) is adopted from an open-source implementation (Xiao, 2018). The implementations of other methods are downloaded from the original paper (Geng and Ji, 2013; Conneau et al., 2017). For EDL-CNN, we use the network architecture in the original paper, and set the learning rate 0.1, epoch number 25 (Kim, 2014). In the same experiment setting with (Zhang et al., 2018), namely, 90% training data and 10% testing data of SemEval 2007, we can obtain 0.344 Euclidean$^{\downarrow}$, 0.330 Sørensen$^{\downarrow}$, 0.344 Squared$\chi^{2\downarrow}$, 0.437 K-L$^{\downarrow}$, 0.853 Fidelity$^{\uparrow}$, 0.670 Intersection$^{\uparrow}$, which are slightly better than the results reported in (Zhang et al., 2018). For fair comparison, we use the default parameters of all other methods, which are reported to get the best performance. For EDL-Meta, we empirically set $\alpha = 0.01$, $\beta = 0.1$, $L = 5$, $H = 5$. $\gamma$ is set to be 0.7 according to (Zhang et al., 2018), and we do not further tune it. 1000 epoches are used to train the meta-learner for both datasets. To find optimal $K$ and $R$, we run a 5-fold cross validation grid search, and set $K = 20$, $R = 5$. The alternating least squares method is used to calculate the tensor decomposition, and we use the implementation in the Matlab tensor toolbox [4]. The k-d tree method is used to find KNNs of a sample efficiently. We run all experiments on a desktop with Intel(R) Core(TM) i9-7900X CPU 3.30GHz, 64GB RAM, Nvidia GeForce GTX 1080 Ti (×2).

## 5.2 Results

The comparison results of SemEval 2007 Task 14 are shown in Table 1. We can see that EDL-Meta shows better results than other methods on all metrics. In particular, EDL-Meta can usually outper-

[2] https://pytorch.org

[3] https://radimrehurek.com/gensim

[4] http://www.tensortoolbox.org

| Method | Euclidean$^{\downarrow}$ | Sørensen$^{\downarrow}$ | Squared$\chi^{2\downarrow}$ | K-L$^{\downarrow}$ | Fidelity$^{\uparrow}$ | Intersection$^{\uparrow}$ |
|---|---|---|---|---|---|---|
| doc2vec | 0.3987 | 0.3795 | 0.4563 | 0.5231 | 0.8279 | 0.6205 |
| InferSent | 0.3993 | 0.3799 | 0.4567 | 0.5228 | 0.8277 | 0.6201 |
| random | 0.3966 | 0.3763 | 0.4494 | 0.5179 | 0.8317 | 0.6237 |
| tensor | **0.3949** | **0.3747** | **0.4478** | **0.5148** | **0.8319** | **0.6253** |

Table 3: Comparison results of different embedding methods.

| Meta. | Adap. | Euclidean$^{\downarrow}$ | Sørensen$^{\downarrow}$ | Squared$\chi^{2\downarrow}$ | K-L$^{\downarrow}$ | Fidelity$^{\uparrow}$ | Intersection$^{\uparrow}$ |
|---|---|---|---|---|---|---|---|
| No | No | 0.4450 | 0.4345 | 0.5553 | 0.6105 | 0.7794 | 0.5655 |
| No | Yes | 0.4588 | 0.4489 | 0.5827 | 0.6403 | 0.7680 | 0.5511 |
| Yes | No | 0.3954 | 0.3752 | 0.4482 | 0.5154 | 0.8318 | 0.6248 |
| Yes | Yes | **0.3949** | **0.3747** | **0.4478** | **0.5148** | **0.8319** | **0.6253** |

Table 4: Comparison results of different combinations of training and adapting procedures.

form other methods by about 5% on most metrics. Although EDL-CNN can perform well with more training samples (Zhang et al., 2018), on the small sample, its performance is not as stable as EDL-Meta. Even simple KNN can get slightly better results than EDL-CNN on most metrics. Likewise, other methods cannot get stable results on all metrics. This indicates that EDL-Meta is more suitable for small sample EDLs.

## 5.3 Discussion

### 5.3.1 Low-rank Embedding

To investigate the effectiveness of the tensor decomposition method, we use doc2vec (Le and Mikolov, 2014) and InferSent (Conneau et al., 2017) to train sentence vectors on $\mathcal{D}$, and find KNNs of a sample by cosine similarity, denoted as doc2vec and InferSent, respectively. In addition, we also use randomly selected K-neighbors to train and test EDL-Meta, denoted as random. We denote our tensor decomposition method as tensor. The results are shown in Table 3. Compared to doc2vec, InferSent, and random, our method can get better results on all metrics. The doc2vec and InferSent embedding methods show a slightly lower performance, even worse than the random version. doc2vec usually assumes a large pool of training data (Le and Mikolov, 2014), while InferSent also does not work well for transferring to a task with a small training sample. The tensor decomposition method is more suitable for small sample learning.

### 5.3.2 Meta-Training and Adapting

To investigate the effect of the proposed meta-training and adapting procedures, we conduct another experiment. In particular, with the same experiment setting, we replace the meta-training procedure (denoted as Meta.) with a normal batch training (batch size 50) and the adaption procedure (denoted as Adap.) with a normal evaluation procedure, respectively. We can then get four versions of EDL-Meta, namely, *without Meta. + without Adap.*, *without Meta. + with Adap.*, *with Meta. + without Adap.*, and *with Meta. + with Adap*. The first one is EDL-CNN in particular, and the last one is the normal EDL-Meta. In this way, we can further examine the importance of the proposed meta-training and adapting procedures. The results are shown in Table 4. We can see that with both the meta-training and adapting procedures, EDL-Meta performs more stably than other methods. In addition, the meta-training procedure plays a crucial role on boosting the performance, as the results of the third version are similar to the fourth version. Finally, without meta-training, only using

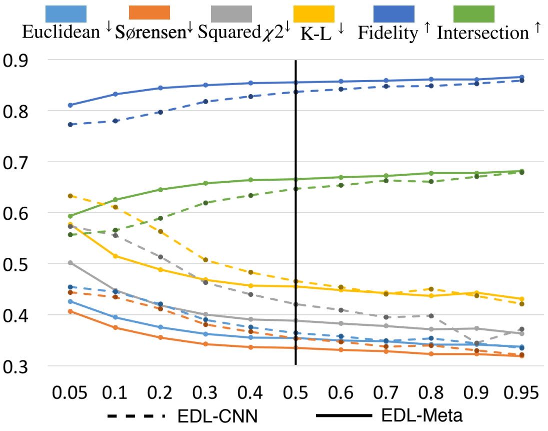

Figure 3: The performance of EDL-CNN and EDL-Meta on different training percentages. The horizontal axis denotes different training percentages, and the vertical axis denotes the performance on different metrics.

adapting is meaningless, as the performance of the second version decreases a lot.

### 5.3.3 Label Percentage

In the same experiment setting, we further investigate the influence of different training/testing partition. We compare the performance of EDL-CNN and EDL-Meta under different training/testing partition. The results are shown in Figure 3. We can find that on small training sample, there is a clear gap between EDL-CNN and EDL-Meta. But with more training sample (>50%), EDL-CNN and EDL-Meta converge to similar results. Therefore, it is likely that the learning ability of EDL-Meta is limited by its basic learner. Further experiments on different learners deserve to investigate.

## 6 Conclusion

In this paper, we propose an efficient meta-learning approach to learn text emotion distribution from a small sample. In addition, to find the K-nearest semantically similar neighbors, we propose to learn sentence embedding by tensor decomposition. Experiments on SemEval 2007 Task 14 shows that the proposed approach outperforms existing EDL methods on small sample emotion distribution learning. We further investigate the low-rank embedding method, the meta-training procedure, and the adaption procedure. We find that leveraging nearest semantically similar neighbors is an effective way for small sample EDLs. We also find that the meta-training procedure can be a good alternative to the normal batch training procedure, especially for small sample EDLs.

## Acknowledgments

The authors would like to thank Lanqing Xue, Zhenhui Peng, and Prof. Evangelos E. Papalexakis for insightful discussions, as well as all anonymous reviewers for constructive comments. This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China under Grant No.: C6030-18G.

## References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. EmoNet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. 1991. Learning a synaptic learning rule. In *International Joint Conference on Neural Networks (IJCNN)*, page 969.

Lea Canales and Patricio Martínez-Barco. 2014. Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, Quito, Ecuador. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Ahmed Fadhil and Silvia Gabrielli. 2017. Addressing challenges in promoting healthy lifestyles: The AI-chatbot approach. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 261–265. ACM.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia. PMLR.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 2019. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1920–1930, Long Beach, California, USA. PMLR.

Xin Geng and Rongzi Ji. 2013. Label distribution learning. *Proceedings of the 13th IEEE International Conference on Data Mining Workshops*, pages 377–383.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods* *in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Ekta Gujral, Ravdeep Pasricha, and Evangelos E. Papalexakis. 2018. Sambaten: Sampling-based batch incremental tensor decomposition. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 387–395.

Seyedmehdi Hosseinimotlagh and Evangelos E Papalexakis. 2018. Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. In *WSDM 2018 Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*.

Kyle Hsu, Sergey Levine, and Chelsea Finn. 2019. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 732–738, New Orleans, Louisiana. Association for Computational Linguistics.

Mark Johnson, Peter Anderson, Mark Dras, and Mark Steedman. 2018. Predicting accuracy on large datasets from smaller pilot data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 450–455, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2012. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5:1–5:8, New York, NY, USA. ACM.

Yanghui Rao, Haoran Xie, Jun Li, Fengmei Jin, Fu Lee Wang, and Qing Li. 2016. Social emotion classification of short text via topic-level maximum entropy model. In *Information & Management*, volume 53, pages 978 – 986.

Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108, Lisboa, Portugal. Association for Computational Linguistics.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1842–1850. JMLR.org.

Jun Shu, Zongben Xu, and Deyu Meng. 2018. Small sample learning in big data era. *arXiv:1808.04572*.

Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.*, 65(13):3551–3582.

Jacopo Staiano and Marco Guerini. 2014. Depeche mood: a lexicon for emotion analysis from crowd annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 427–433, Baltimore, Maryland. Association for Computational Linguistics.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 70–74.

Carlo Strapparava and Alessandro Valitutti. 2004. WordNet affect: an affective extension of WordNet. In *Proceedings of the Fourth International*

*Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Sebastian Thrun and Lorien Pratt, editors. 1998. *Learning to Learn*. Kluwer Academic Publishers, Norwell, MA, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95.

Ekaterina P. Volkova, Betty J. Mohler, Detmar Meurers, Dale Gerdemann, and Heinrich H. Bülthoff. 2010. Emotional perception of fairy tales: Achieving agreement in emotion annotation of text. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 98–106, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv.*, 50(2):25:1–25:33.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Jianfei Yu, Luis Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. 2018. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1097–1102, Brussels, Belgium. Association for Computational Linguistics.

Yuxiang Zhang, Jiamei Fu, Dongyu She, Ying Zhang, Senzhang Wang, and Jufeng Yang. 2018. Text emotion distribution learning via multi-task convolutional neural network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4595–4601. International Joint Conferences on Artificial Intelligence Organization.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Austin, Texas. Association for Computational Linguistics.