一种基于深度学习的应用于推特文本的多标签情感分类算法

摘要

目前，大多人使用网络社交媒体如推特或脸书来分享他们的情感以及想法。识别和分析这些在社交平台表达的情感能有利于商业、公共卫生、社会福利等诸多领域的应用。大多数过去在情感分析的研究只关注单标签分类而忽略了一个句子中共表达的多种情感标签。在本论文中描述了一种用于推特数据的，新颖的基于深度学习系统来解决多标签情绪分类问题的开发。我们提出了一种新奇的方法来将多标签情绪分类问题转为一个二元分类问题然后利用深度学习方法来解决该转化问题。我们的系统的性能表现比当前最好的系统优秀，在具有挑战性的 SemEval2018 Task1: Affect in Tweet 任务中实现了 0.59 的准确性分数。

关键词：观点挖掘；情绪分析；情感分类；深度学习；推特

1. 简介

感情是人们如何感觉和思考的关键。网络社交媒体，如推特和脸书，已经改变了交流的语言。目前，人们可以在短文本中交流不同话题的事实、观点、情感和情感强度。分析社交媒体内容中表达的情感吸引了自然语言处理研究领域的学者们。它在商业、公共卫生、社会福利等方面有着广泛的应用。例如，它可以用于公共卫生，公众的政治倾向检测，品牌管理，股市监测。情绪分析是确定人们对目标或话题的态度的任务。这种态度可以是极性的(积极的或消极的)或是一种情绪状态，如喜悦、愤怒或悲伤。

近些日子里，多标签分类问题因其在文本分类、场景和视频分类、生物信息学等领域的广泛应用而引起了可观的关注。与传统的单标签分类问题(即多类或二分类问题)不同，在单标签分类问题中，一个实例只与有限标签集合中的一个标签相关，而在多标签分类问题中，一个实例与一个标签子集相关。

过去的情感和情绪分析工作大多只关注于单标签分类。因此，在本文中，我们专注于多标签情绪分类任务，目的在于开发一个自动系统，以确定文本中是否存在 11 种情绪中的任何一种或一种以上:如图 1 所示的 8 种普鲁切克类别(喜悦、悲伤、愤怒、恐惧、信任、厌恶、惊讶和期待)，以及乐观、悲观和爱。

众多解决多标签分类问题最常用的方法之一是问题转换。该方法将多标记问题转化为一个或多个单标记(即二元或多类)问题。具体来说，单标签分类器进行学习和被应用与改进;然后，将该分类器的预测转化为多标签预测。

在多标签研究的文献中，已经提出了不同的转换方法。最常见的方法是二元相关法。二元相关法的思想简单直观。将多标签问题转化为多个二值问题，每个标签对应一个问题。然后，训练一个独立的二分类器来预测其中一个标签的相关性。虽然二元相关在文献中很流行，但由于其简单性，无法直接建模标签之间可能存在的相关性。然而，它对过拟合标签组合有很强的抵抗力，因为它不期望样本与先前观察到的标签组合有相关联。

在本文中，针对多标签分类问题，我们提出了一种新的变换方法 xy-pair-set。与二元相关方法不同，我们的方法将问题转化为仅一个二元分类问题。此外，利用深度学习模型的成功，特别是 word2vemethods 的系列家族和循环神经网络和注意力模型，开发了一个解决转

换后的二分类问题的系统。该系统的关键部分是嵌入模块，它使用三个嵌入模型和一个注意力函数来对输入和标签之间的关系进行建模。

总结的来说，这工作可分为四个部分的贡献

- 我们针对多标签分类问题，提出了一种新的转换机制。
- 我们提出一种新的基于注意力机制的深度学习系统，称为二值神经网络(BNet)，基于新的转换方法工作。该系统是一个数据驱动的、基于神经的端到端模型，不依赖于词性标记器和情感或情感词典等外部资源。
- 在 SemEval-2018 Task1: Affect in Tweets 的具有挑战性的多标签情绪分类数据集上评估了所提出的系统。
- 实验结果表明，该系统的性能优于目前最先进的系统。

本文其余部分的结构如下。在第 2 节中，我们概述了多标签问题转换方法和对于 Twitter 文本的情绪极其情绪分析的相关工作。在第 3 节中，我们将详细解释这种方法。在第 4 节中，我们报告了实验结果。在第 5 节中，提出了结论和未来的工作。

2. 相关工作

在本节中，我们回顾了与这项工作最流行的的相关工作。在 2.1 节我们总结了最普遍的多标签问题转换方法。在 2.2 节，给出了关于 Twitter 上多标签情感分类问题的最新研究综述。

2.1 问题转换方法

设 X = {x1, x2，…xn}为所有实例的集合，Y = {y1, y2，…， ym}是所有标签的集合。我们可以定义数据集:

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

在这个表达式中，D 被称为有监督的多标签数据集。

多标签分类任务很有挑战性，因为标签集的数量会随着类别标签数量的增加而呈指数级增长。解决该问题的一种常见方法是将该问题转化为传统的分类问题。该思想是通过利用标签相关性来简化学习过程。基于相关性的顺序，我们可以将现有的转换方法分组为三种方法，即一阶方法、二阶方法和高阶方法。

一阶方法将问题分解为多个独立的二分类问题。在这种情况下，为每个可能的类别学习一个二分类器，忽略了其他标签同时存在的情况。因此，所需的独立二分类器的数量与标签的数量相同。对于每个多标签训练样例，我们构建一个二分类训练集 Dk 如下:如果 yk∈ˆYi，则 xi 为一个正例，否则为一个反例。在第一种情况下，我们将得到一个形式为(xi, 1)∈Dk 的训练样本，在第二种情况下，训练样本为(xi, 0)∈Dk。因此，对于所有标签{y1, y2，…， ym}∈Y, m 个训练集{D1, D2，…， Dm}被构建出。在此基础上，对于每个训练集 Dk，可以使用流行的学习技术如 AdaBoost、k 近邻、决策树、随机森林等学习一个二分类器。一阶方法的主要优点是概念简单和高效。然而，这些方法可能不太有效，因为它们忽略了标签的相关性。

二阶方法试图通过利用标签之间的成对关系来解决标签相关性建模的不足。考虑成对关系的一种方法是为每一对标签训练一个二分类器。虽然二阶方法在一些领域表现良好，但在训练器数量方面比一阶方法更复杂。它们的复杂度是二次的，因为所需的分类器数量是从 m 个标签中取出 2 个的组合数。此外，在现实世界的应用中，标签相关性可能更复杂，超越二阶。

高阶方法通过探索标记之间的高阶关系来解决多标记学习问题。这可以通过假设线性组

合、非线性映射或整个标记空间上的共享子空间来实现。尽管高阶方法比一阶和二阶方法具有更强的相关性建模能力，但这些方法的计算要求高，可扩展性差。

我们在 3.1 节中展示的转换机制像一阶方法一样简单，如果满足 3.1 节中详细的一些要求，可以隐式地对标签之间的高阶关系进行建模。该算法只需要一个二分类器，并且训练样本的数量随着样例数量和标签数量的增加呈多项式增长。如果多标签练数据集中的训练样本数为 n，标签数为 m，则转换后的二元训练集中的训练样本数为 n × m。

2.2 推特中的情感分类

对于传统情感分类和多标签情感分类问题，已有各种各样机器学习方法被提出。现有的系统大多将该问题作为文本分类问题来解决。有监督的分类器是在一组标注的语料库上使用另一组手工设计的特征进行训练的。此类模型的成功归于两个主要因素:大量的标记数据和人为精心设计的巧妙能够区分样本的一组特征。使用这种方法，大多数研究都集中在精心设计一组有效的特征以获得良好的分类性能。其思想是找到一组信息丰富的特征来反映文本中表达的情感或情感。词袋模型(Bag-of-Words, BoW)及其变体 n-grams 是大多数文本分类问题和情感分析中使用的表示方法。不同的研究将 BoW 特征与词性标记、从词典中提取的情感和情感信息、统计信息、词的形状等其他特征相结合来丰富文本表示。

尽管 BoW 在大多数文本分类系统中是一种流行的方法，但它存在一些缺陷。首先，它忽略了语序。这意味着两个文档可能具有相同或非常接近的表示，只要它们具有相同的单词，即使它们携带不同的含义。n-gram 方法解决了 BoW 方法在长度为 n 的语境中考虑词序的缺点，但存在稀疏性和高维性的问题。其次，BoW 难以对单词语义进行建模。例如，beautiful、wonderful 和 view 在 BoW 中有相等的距离，其中 beautiful 比 view 在语义空间中更接近 wonderful。

情绪和情感词典在构建高效的情感分析系统中起着至关重要的作用。然而，创建这样的词典是困难的。另外，除了寻找最佳的统计特征集之外，寻找最佳的词典组合也是一项耗时的任务。

目前，深度学习模型被用于多种开发端到端的系统的任务中，包括语音识别、文本分类和图像分类。研究表明，这样的系统可以自动地从原始数据中提取高层次的特征。

Baziotis 等人（SemEval-2018 Task1: Affect in Tweets 的多标签情感分类任务冠军）开发了一种带有深度注意力机制的双向长短期记忆(LSTM)模型。他们从 80 万个单词来自 5.5 亿条推文的数据集训练了一个 word2vec 模型。SemEval 排行榜的第二名使用注意力机制训练了一个单词级的双向 LSTM，它还在集成中包含了非深度学习的特征。Ji Ho Park et al.训练了两个模型来解决这个问题:正则化线性回归和逻辑回归分类器链。他们试图利用标签的相关性来进行多标签分类。对于第一个模型，他们将多标签分类问题建模为一个以标签距离为正则项的线性回归问题。在他们的工作中，逻辑回归分类器链方法被用于捕获情感标签相关性。其思想是将多标签问题视为一系列二分类问题，将上一个分类器的预测作为下一个分类器的额外输入。

本文利用基于深度学习的方法开发了一个系统，可以提取推特文本的高级表示，并对标签之间的隐式高阶关系进行建模。我们将提出的系统与提出的转换方法一起训练了一个可以解决推特文本数据的多标签情感分类问题的函数。下一节将解释我们提出的系统的细节。

3. 方法论

在本节中展示了我们工作的方法。首先，我们在 3.1 节中解释了所提出的转换方法 xy-pair-set。之后，我们将在第 3.2 节描述所提议的系统。

3.1 xy-Pair-Set：问题转换

所提出的问题转化方法 xy-pair-set 将多标签分类数据集 D 转化为有监督的二元数据集^D 如下：

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

算法 1 解释了所提出的转换方法的实现。它以一个多标签数据集 D(等式(1))和一组标签 Y 作为输入，并返回一个转换后的二值数据集。我们接下来展示一个说明性的例子。

让 X = {x1, x2}, Y = {a, b, c}和 D = {(x1, {a, c}), (x2, {b})}。二元相关性转换方法的输出是一组三个独立的二元数据集，每个标签一个。也就是说,Da = {(x1, 1), (x2 0)}, Db = {(x1, 0), (x2, 1)}, Dc = {(x1, 1), (x2 0)}。相比之下，我们转换方法的输出是一个单一的二元数据集ˆD = {((x1, a)， 1)，((x1, b)， 0)，((x1, c)， 1)，((x2, a)， 0)，((x2, b)， 1)，((x2, c)， 0)}。

这种情况下的任务与传统的监督二分类不同，是用来开发一个学习算法来学习一个函数 g: X × Y→{0,1}。这种算法的成功基于三个条件:(1)将实例 x∈X 表示为高维的向量 Vx 的编码方法，(2)将 y∈Y 的标签编码为向量 Vy 的方法，(3)表示实例 x 与标签 y 之间关系的方法。这三个条件使得 g 能够捕获输入到标签和标签到标签的关系。本文利用深度学习模型的成功，来满足上述三个要求。我们凭经验展示了我们的系统在这些条件下的成功，如第 4.6 节和 4.7 节所述。

3.2 BNet：系统描述

本小节解释了所提议的系统来解决上面提到的转换二进制问题。图 2 显示了系统架构的图形化描述。它由嵌入模块、编码模块和分类模块三部分组成。我们将在下面详细解释它们。

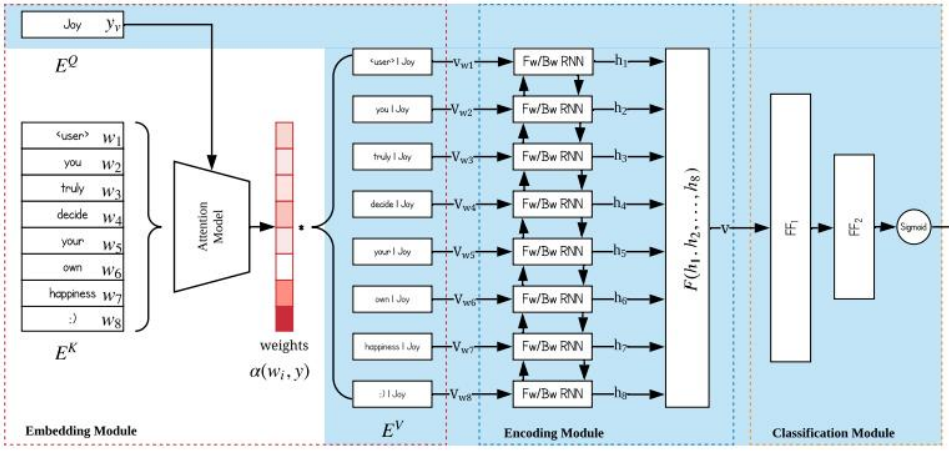

Figure 2. An illustration of the proposed system. Shaded parts are trainable. Fw and Bw refer to the Forward and Backward cells respectively, and FF means Feed-Forward layer.

3.2.1 嵌入模块

设(W, y)为系统的一对输入，其中 W = {w1, w2，…， wl}是推特文本中单词的集合，y 是情感对应的标签。嵌入模块的目标是用向量 vwi 表示每个单词 wi，用向量 vy 表示标签。

我们的嵌入模块可以看作是一个函数，它将查询和一组键值对映射到输出，其中查询、键、值和输出都是向量。输出被计算为值的加权和，其中分配给每个值的权重是由查询与相应键的兼容性函数计算的。查询是可训练的标签嵌入，E Q(y);关键字是预训练的词嵌入，E K(wi)∀wi∈W;其中的值就是可训练的词嵌入，E V(wi)∀wi∈W。

如图 2 所示，我们将 E Q 和 E K 的输出作为注意力模型的输入，来寻找标签 y 和输入推文的单词 W 之间的对齐，即权重α。这一步对输入和标签之间的关系进行建模。一旦获得权重，我们就将来自嵌入 E V 的每个单词的向量乘以其相应的权重。综上所述，单词 wi∈W 的最终表示如下:

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

函数α是一个基于注意力的模型，它根据单词 wi 和标签 y 之间的语义相似度来发现它们之间的关系强度。即α(wi, y)是一个基于 wi 与 y 之间的距离Δ(wi, y)的值，如下所示:

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

这里，Δ(wi, y)是一个标量分数，表示单词 wi 和标签 y 之间的相似度:

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

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

值得注意的是，α(wi, y)∈[0,1]，且:

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

3.2.2 编码模块

编码模块的目标是映射单词表示序列{vw1, vw2，…， VWl}，从嵌入模块获得单个实值稠密向量。在这项工作中，我们使用循环神经网络(RNN)来设计编码器。RNN 从第一个符号 vw1 到最后一个符号 vwl，按向前的方向(从左到右)读取输入的向量序列。因此，它按时间顺序处理序列，忽略未来的上下文。然而，对于序列上的许多任务，能够访问未来和过去的信息是有益的。例如，在文本处理中，决策通常是在整个句子已知后做出的。双向循环神经网络(BiRNN)变体提出了一种基于过去和未来信息进行预测的解决方案。

BiRNN 由正向→ ϕ 和逆向← ϕ 两部分组成。第一个按前向方向读取输入序列，并产生一个前向隐藏状态序列(→h1，…，→hl)，而前者以相反的顺序读取序列(vwl，…， vw1)，从而产生一个向后隐藏状态序列(←hl，…，←h1)。

我们通过连接对应的前向隐藏状态−→ht 和后向隐藏状态←−ht 得到每个单词的 vwt 表示。下面的公式说明了其主要思想:

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

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

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

最终输入的序列的表示为：

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

我们简单的选择 F 作为最后的隐藏状态(即 F({h1, h2，…， hn}) = hn)。

在这项工作中，我们使用两个门控循环单元(Gated Recurrent Units, gru)，一个是→ϕ，另一个是←ϕ。这种 RNN 被设计为具有更持久的记忆，使它们在捕获序列元素之间的长期依赖关系方面非常有用。图 3 显示了门控循环单元的图形描述。

GRU 具有复位(rt)门和更新(zt)门。如果前者发现过去的隐藏状态 ht - 1 与新状态的计算无关，则它可以完全忘记过去的隐藏状态 ht - 1，而后者负责确定有多少 ht - 1 应该被带入下一个状态 ht。

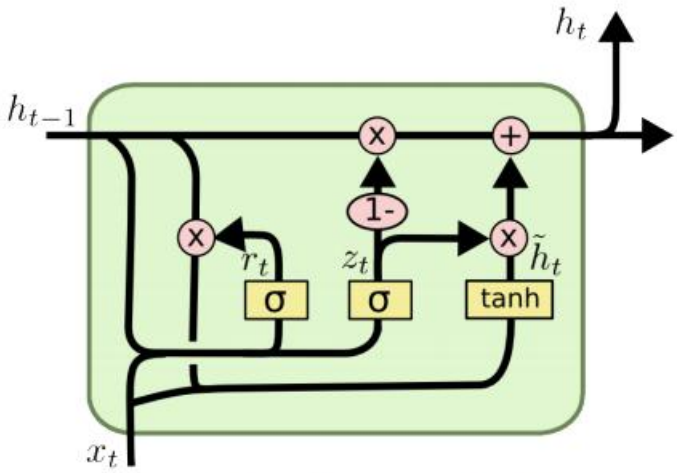

Figure 3. Gated Recurrent Unit (GRU).

GRU 的输出 ht 依赖于输入 xt 和前一个状态 ht−1，其计算如下:

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

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

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

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

在这些表达式中，rt 和 zt 表示重置门和更新门，~ht 是候选输出状态，ht 是时刻 t 的实际输出状态。符号〇代表元素乘法; σ 是一个 sigmoid 函数;以及;表示向量拼接操作。Wr、Wz、Wh∈Rdh×(d+dh)和 br、bz、bh∈Rdh 是复位门和更新门的参数，其中 dh 是隐藏状态的维数，d 是输入向量的维数。

3.2.3 分类模型

我们的分类器由两个激活函数为 ReLU 的向前反馈层跟着一个 Sigmoid 单元组成。

4. 实验和结果

在本节中，我们首先描述实验细节，然后描述数据集和我们使用的预处理。随后，我们介绍了与我们的系统进行比较的最先进的系统，最后，我们报告了实证验证，证明了我们的系统的有效性。

4.1 实验细节

表 1 显示了我们的系统的超参数，使用 Adam 算法进行训练，学习率为 0.0001，β 1 = 0.5，小批量大小为 32，以最小化二元交叉熵损失函数:

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

其中，g(xi, yi)为预测值， Φi 为实值， θ 为模型参数。
通过贝叶斯优化得到系统的超参数。我们使用开发集作为验证集来微调这些参数。

Table 1. Hyperparameters of our system.

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

4.2 数据集

在实验中，我们使用 SemEval-2018 Task1: Affect In Tweets 的多标签情感分类数据集。它包含 10,983 个样本，分为三部分:训练集(6838 个样本)、验证集(886 个样本)和测试集(3259

个样本)。有关数据集的更多细节，请查阅本文参考文献 38。我们在训练集上训练了我们的系统，并使用开发集微调了所提出系统的参数。我们对数据集中的每条推文进行了如下预处理:

• 标记化(Tokenization):我们使用了一个广泛的正则表达式列表来识别推文中包含的以下元信息:Twitter 标记、表情符号、表情符号、日期、时间、货币、缩略词、话题标签、用户提及、url 和强调的单词。

• 分词完成后，我们将单词小写，并对识别出的单词进行规范化。例如，URLs 被替换为标记词“<URL>”，用户的指代被替换为标记“<USER>”。这一步有助于在不丢失信息的情况下减少词汇表的大小。

4.3 与其他系统的对比

将提出的系统与用于多标签情感分类任务的最新系统进行了比较，包括:

• SVM-unigrams:仅使用单词 unigrams 作为特征训练的基线支持向量机系统。

• NTUA-SLP: SemEval-2018 任务 1:E-cchallenge 获奖团队提交的系统。

• TCS:第二名提交的系统。

• PlusEmo2Vec:第三名提交的系统。

• Transformer:由 NVIDIA AI lab 开发的大型预训练语言模型的深度学习系统。

4.4 评价指标

我们在 E-c 子任务中使用了多标签准确率(或 Jaccard 指数)，这是 SemEval-2018 任务 1:影响推文的组织者使用的官方竞赛指标，可以定义为预测标签集和黄金标签集的交集的大小除以它们的并集的大小。

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

在这个表达式中，Gt 是推文 t 的黄金标签集合，Pt 是推文 t 的预测标签集合，T 是推文集合。此外，我们还使用微平均 F -分数和宏平均 F -分数。

设#c(l)表示正确分配给标签 l 的样本数量，#p(l)表示分配给 l 的样本数量，#(l)表示 l 中实际样本的数量。微平均 F1 -分数的计算方法如下:

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

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

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

因此，Pmicro 是微平均精确率分数，Rmicro 是微平均召回率分数。

让 Pl、Rl 和 Fl 表示标签 l 的精度分数、召回率分数和 F1 -分数。宏平均的 F1 -分数计算如下:

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

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

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

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

4.5 结果

我们将系统的预测提交给 SemEval Task1:E-C 挑战。结果是由组织者在黄金测试集上计算的，在这中我们无法获得黄金标签。

表 2 显示了我们的系统结果和对比模型的结果(从它们的相关论文中获得)。从报告的结果可以看出，该系统在所有最先进的系统中取得了最高的 Jaccard 指数准确性和宏平均 F1 分数，在微平均 F1 分数上具有竞争力，但略低。

为了更深入地了解我们系统的性能，我们计算了每个标签的精度分数、召回率分数和 F1 分数。此分析的结果如图 4 所示。我们发现我们的系统在“快乐”标签上表现最好，其次是“愤怒”、“恐惧”、“厌恶”和“乐观”标签。这些标签的 F1 值均在 70%以上。在“信任”、“惊讶”、“期待”和“悲观”标签上表现最差。在大多数情况下，我们的系统给出的召回率分数高于精确率分数。系统似乎对“信任”、“惊讶”、“预期”和“悲观”(即，与数量少的样本相关的标签）不敏感。这可以归因于这些情绪的训练样本数量较少以及 Out-Of-Vocabulary(OOV)问题。

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

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

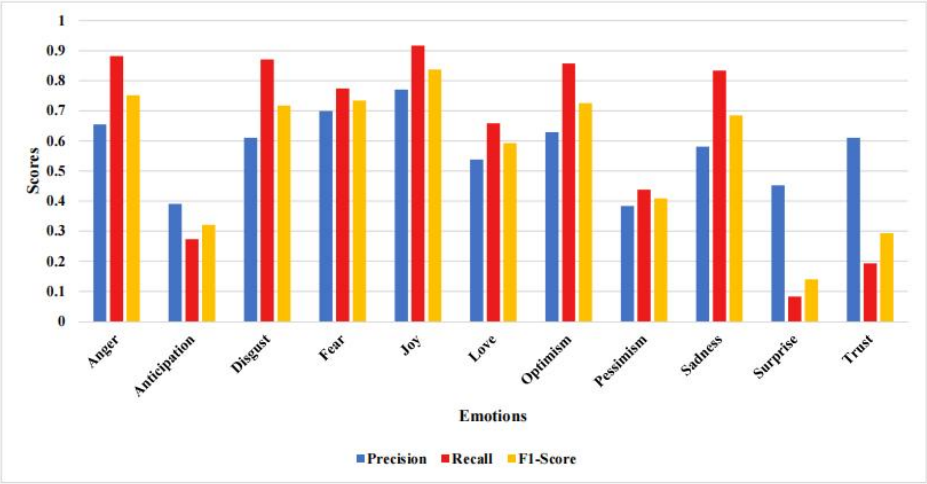

Figure 4. Performance analysis.

4.6 注意力可视化

我们将注意力权重可视化，以更好地了解我们系统的性能。图 5-8 描述了结果，显示了验证集中前四条示例推文的注意力权重的热图。颜色强度指的是注意力模型赋予每个单词的权重。它代表了单词和情感之间的关系强度，反映了这个单词在最终预测中的重要程度。我们可以看到，在推文没有分配到情感的情况下，注意力模型给了常见单词(如停用词)重要的权重;例如，图 5 中的单词“for”和图 7 中的单词“this”以及图 8 中的标记“<user>”。此外，它还对与情绪相关的单词和表情符号给予很高的权重(例如，“欢呼”和“棒极了”表示喜悦，“生日”表示爱，等等)。一个有趣的发现是，当表情符号出现时，如果它们与情绪相关，那么它们几乎总是被选择为重要的表情。例如，我们可以在图 7 中看到，悲伤情绪在很大程度上依赖于表情符号。我们还发现，仅考虑一个单词来建模推文和情感之间的关系是不够的。在某些情况下，一个单词的情感可能会根据上下文发生翻转。例如，以下面的推文为例:“当#富有成效(做需要做的事情)时，#焦虑水平降低，#爱的水平增加。#个人性取向”，“焦虑”一词与恐惧情绪高度相关，但在这种情况下，它显示的是乐观和信任情绪。然而，我们的系统错误地将这个例子与恐惧情绪联系起来。

	im	clapping	and	cheering	for	both	teams	.	..1	..2
Anger	0.43	0.10	0.18	0.01	0.00	0.01	0.02	0.08	0.08	0.08
Anticipation	0.03	0.02	0.05	0.04	0.28	0.07	0.19	0.11	0.11	0.11
Disgust	0.15	0.08	0.26	0.01	0.01	0.02	0.04	0.14	0.14	0.14
Fear	0.01	0.01	0.02	0.00	0.23	0.01	0.00	0.24	0.24	0.24
Joy	0.01	0.07	0.00	0.74	0.13	0.04	0.01	0.00	0.00	0.00
Love	0.00	0.00	0.00	0.01	0.87	0.00	0.00	0.01	0.01	0.01
Optimism	0.01	0.01	0.02	0.13	0.50	0.18	0.09	0.02	0.02	0.02
Pessimism	0.01	0.01	0.02	0.01	0.30	0.01	0.03	0.20	0.20	0.20
Sadness	0.03	0.02	0.03	0.02	0.21	0.02	0.02	0.21	0.21	0.21
Surprise	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Trust	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

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

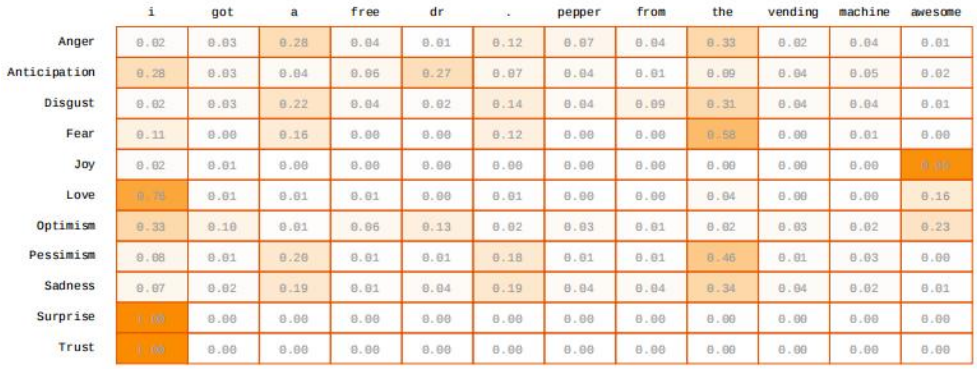

Figure 6. Attention visualization example. Golden labels are {*joy*, *surprise*} and predicted labels are {*joy (0.97)*, *optimism (0.87)*}.

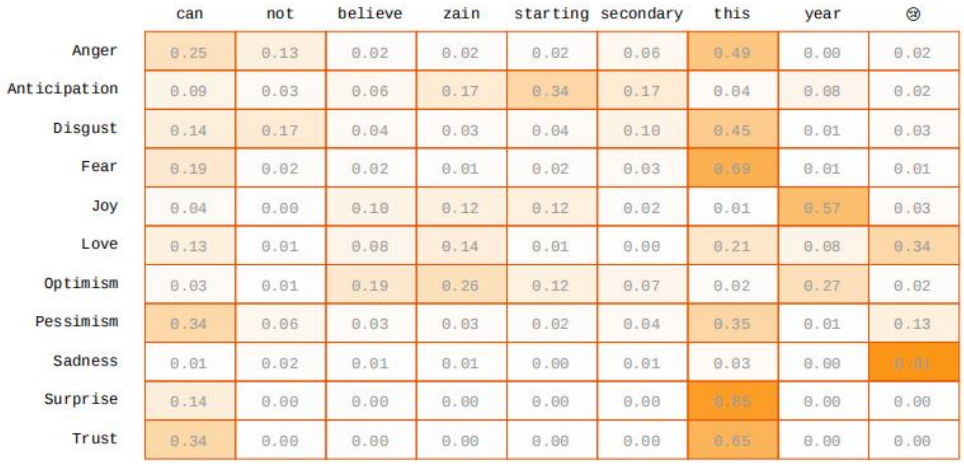

Figure 7. Attention visualization example. Golden labels are {*sadness*, *surprise*} and predicted labels are {*love (0.74)*, *sadness (0.98)*}.

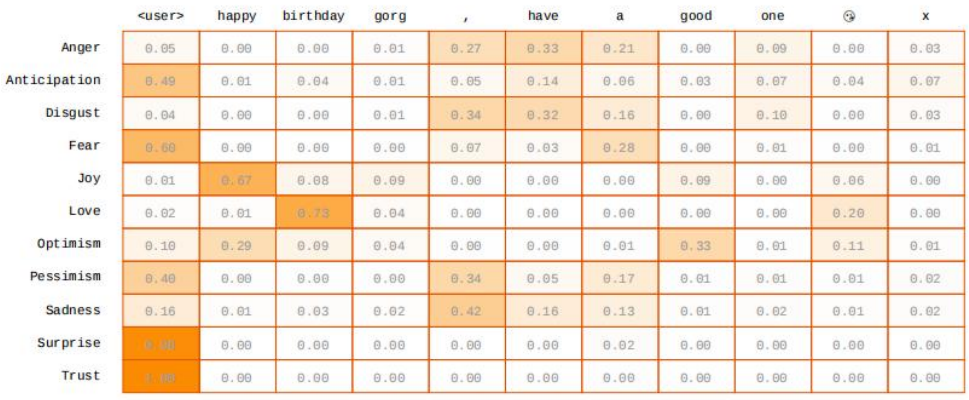

Figure 8. Attention visualization example. Golden labels are {*joy*, *love*, *optimism*} and predicted labels are {*joy (0.98)*, *love (0.91) optimism (0.95)*}.

4.7 相关性分析

图 9 显示了验证集中情绪标签的相关性分析。图中的每个单元表示每对情感标签的相关

性得分。报告的数值显示了令人兴奋的发现。该系统捕获了情感标签之间的关系。预测标签的相关分数几乎与真实值相同。惊喜和信任情绪是一个例外。我们的系统未能捕捉到这两种情绪和输入或其他情绪之间的关系。将这种明显的相关性缺乏归因于这两种情绪的训练样本数量较少。

此外，“快乐”和“乐观”等相关情绪之间总是呈正相关的(真实标签和预测标签的分数为 0.74)。另一方面，我们可以看到，“愤怒”和“爱”等不相关的情绪之间存在负相关。得分分别为- 0.27 和- 0.3。

这一结果进一步加强了我们的假设，即所提出的系统能够隐式地对情绪标签之间的关系进行建模。

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

(a) The ground-truth labels.

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

(b) The predicted labels.

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

5. 总结

在这项工作中，我们提出了一种新的多标签情感分类方法。首先，提出一种转换方法将问题转化为单一的二分类问题。之后，我们开发了一个基于深度学习的系统来解决转换后的问题。该系统的关键部分是嵌入模块，它使用了三个嵌入模型和一个注意力函数。该系统的表现超过了最先进的系统，在具有挑战性的 SemEval2018 任务 1:E-c 多标签情感分类问题上取得了 0.59 的 Jaccard(即多标签准确率)分数。

我们发现注意力函数可以对输入单词和标签之间的关系进行建模，这有助于提高系统的性能。通过将注意力权重可视化并对其进行分析，表明了该系统是可解释的。但是，发现了一些局限性。我们的系统没有对短语和标签之间的关系进行建模。短语在决定分配给一条推文的最合适的情绪集方面起着关键作用。例如，一个反映“悲伤”的情感词可以在否定短语或上下文中翻转。因此，在未来的工作中，我们计划解决这个缺点。一种可能的解决方案是调整注意力函数来建模不同 n-gram 标记和标签之间的关系。结构化注意力网络也可以适应并用于解决这个问题。

此外，我们计划通过实验不同的想法，如使用数据增强来丰富训练数据或使用迁移学习，来开发一个非极端的系统，使得在所有情感标签上表现健壮和平等。