Sequence Generation Model for MLC

1. 研究背景

- **关于MLC**
  - Binary Classification problem: 判断题，example: positive or negative comments?
  - Multi-Class Classification (MCC) problem: 单选题，example: is this a dog, cat, rabbit, or horse pictures?
  - Multi-Label Classification (MLC) problem: 多选题, example labels of comments
    - *"but the staff was so horrible to us"*: `about service`
    - *"the service is great and i love the tomato eggs"*: `about service, about food`
    - *"i really like the smoke duck but the price is high af"*: `about food, about price`

    etc.

  MLC: Assign two or more labels to a sequence of input

- **为什么要设计此实验?**
  - MLC is a very important problem to solve, especially in the field of NLP. Real cases of MLC are text categorization, hashtag recommendation for social media content, labeling article with multiple topics, retrieve important information from a text, etc.
  - 为了完善之前做过的一些解决MLC问题的模型，此文章将对这些模型进行对比：
    - **Binary Relevance**: transform MLC task into small independent binary classification problems. Suppose we have *q* different labels can be assigned to an input. Then we do binary classification for total *q* times to determine whether our input matches with each label. Drawbacks: no correlations between labels.
    - **Classifier Chain**: same as Binary Relevance, but all classifiers are linked in a chain through feature space. feature space具体是什么这个还没弄清楚，但是我估计它的存在使得此方法computationally expensive
    - **Label Powerset**: transform MLC into MCC, where each MCC will try all different subsets from label set. 我觉得此方法也computationally expensive（文章里没讲），因为如果我们的标签集有*n*个元素，那么要进行*2^n*次MCC/?
    - **CNN**: use multiple convolution kernels to extract text features, then give to the linear transformation layer with the sigmoid as the activation function to give the output of probability distribution over the label space. This also ignores the correlation between labels.
    - **CNN-RNN**: utilizes CNN and RNN to capture both global and local textual semantics and model the label correlations. This also ignores the correlation between labels and do not consider differences in the contributions of textual content when predicting labels.
  - 此实验目标：view MLC as a seq2seq problem, generate sequence of labels that are correlated each other. 我的问题是，什么叫做correlated labels？在标签之间有什么重要的关系？为什么要考虑这些关系？

2. 模型设计

以下简单解释文章里设计的模型：

- Encoder
  - The input is a sentence or sequence of words, each word has its one-hot vector representation.
  - Embed (是乘法？) those **one-hot vectors** by an **embedding matrix**, the result is a **dense embedding vector**. 此步骤我估计是把语句里每个词换成词向量/?
  - Use Bidirectional Long-Short Term Memory, takes the **dense embedding vector** as the input, and compute all **hidden states for each word**. What is LSTM?
    - LSTM is kind of RNN model, but it solves gradient vanish and gradient explosion problem.
    - It has long and short term memory value, this value will be adjusted in every iteration.
    - It has forgot gate, input gate, and output gate. These gates will update the long and short term memory value.
    - LSTM uses sigmoid and tanh activation function in its gates.
    - Bidirectional LSTM: read the input from left to right, and right to left. This will help the model to better understand the relationship between following and preceding words.
- Attention

  As the name suggest, "attention" means we only focus on some words in the sentence that will give the most contributions when predicting labels. This step takes our LSTM **hidden states value** as the input, and produce a **context vector**.
- Decoder（？？？）

  Also use LSTM (unidirectional), where each hidden states is computed based on prior hidden states' value and the global embedding of prior iteration's predicted label. 具体的decoder设计以及global embedding我还是不太理解 :(

3. 回答孟学长的问题

- 文章有哪些评价指标 (Evaluation Metrics)？其计算公式？反映了模型的哪些性能?
  - Hamming-loss (文章主要评价指标)

$$\text{Hamming Loss} = \frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L}\left[I(y_j^{(i)} \neq \hat{y}_j^{(i)}\right]$$

    HL is the fraction of wrong predicted labels to the total number of labels (maybe we predict irrelevant labels or the relevant labels are missing).

    我来举个例子来解释。比如说给模型一个句子：`I hate summer in Hangzhou, it's very hot until the sun burns my skin`。假如我们有几个labels可选："positive", "negative", "food", "weather", "disease", "animal", "habit"。

- 模型的输出为：["negative", "weather", "animal"], 这向量是公式里的y^.y^1= negative, y^2 = weather, y^3 = animal.
- 正确的输出为：["negative", "weather", "disease"], 这向是公式里的y, y1 = negative, y2 = weather, y3 = disease.
- 这里有2个误差：模型输出"animal" 以及 模型没输出"disease".
- 总共有7个可选的标签，所以L=7 (also can say we have 7 classes)
- i代表训练数据数目 。因此我们对每个训练数据计算hamming loss再求和。

What i understand from hamming loss: if only part of predicted sequence is wrong, it doesn't mean our prediction is 100% wrong, ONLY PARTIALLY WRONG. Consider this case, suppose we have 2 models which output preds_0 and preds_1:

```
labels = [[0,1,1],[1,1,1],[0,1,0],[1,0,1]]
preds_0 = [[0,1,1],[1,1,0],[1,0,1],[0,1,1]]
preds_1 = [[0,1,1],[1,1,0],[1,1,0],[1,1,1]]
```

If do not use hamming loss, model_0 and model_1 will have the same accuracy: 0.25 (since both of them only guess correct 1 time). But it's not fair, since for 3rd and 4th labels， model_1 predicts closer (hence better) than model_0.

于是，此评价指标会更公平地反映模型的精确度（accuracy）。求学长们对我的理解做个评价~

- Micro-F1 (文章主要评价指标)

  To understand Micro-F1, first look at this table:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

  From now:

  - True Negative = TN
  - True Positive = PN
  - False Negative = FN
  - False Positive = FP

  The terms **micro** means *micro average*, which describe how we count TN, PN, FN, and FP. When we count them, we do not separate each class and calculate them only by checking whether a prediction is right or wrong. Other than *micro average*, we have *macro average*, *samples average*, *weighted average*.

  To understand the terms **F1**, first look at these 2 concepts:

  Precision:

  Of all positive predictions I made, how many of them are truly positive?

$$\text{Precision} = \frac{TP}{TP + FP}$$

Micro-Precision 评价指标也在文章里展示，但不是主要的评价指标。

Recall:

Of all the actual positive examples out there, how many of them did i correctly predict to be positive?

$$\text{Recall} = \frac{TP}{TP + FN}$$

Micro-Recall 评价指标也在文章里展示，但不是主要的评价指标。

F1 is the harmonic mean of precision and recall.

$$F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

对上次例子来讲：

```
labels = [[0,1,1],[1,1,1],[0,1,0],[1,0,1]]
preds_0 = [[0,1,1],[1,1,0],[1,0,1],[0,1,1]]
preds_1 = [[0,1,1],[1,1,0],[1,1,0],[1,1,1]]
```

对模型0进行分析：

- TP = 5, TN = 1, FP = 3, FN = 3
- Precision = 5/(5+3) = 5/8
- Recall = 5/(5+3) = 5/8
- F1 = 5/8

对模型1进行分析：

- TP = 7, TN = 2, FP = 2, FN = 1
- Precision = 7/(7+2) = 7/9
- Recall = 7/(7+1) = 7/8
- F1 = 14/17 (bigger than model_0)

于是，此评价指标也反映了模型的精确度。

问题：

- F1评价为什么要用harmonic mean? 为什么不能用普通的均值或者几何均值?

- Why we only calculate the portion of TP? Why don't we also appreciate TN, since it also true? 换句话说，上面公式里为什么一点也没看到TN？难虽然TN是正确的输出，道我们计算评价指标中是不考虑TN的吗？
- 目前文章里的2个主要评价指标都反映了模型的精确度，但是它们2个区别在哪里？hamming比micro-f1好，还是micro-f1比hamming好？