

基于简化LDSGM模型的隐式篇章关系判别

1120200563 肖良寿 *

December 22, 2022

摘要

隐式篇章关系判别(implicit discourse relation recognition, IDRR)是自然语言理解任务中篇章分析领域的重要理论。本实验借助Changxing Wu等2021年提出的LDSGM模型进行IDRR实验,在模型的Encoder中引入RoBERTa预训练模型进行Label Attentive,在Encoder之后接两个具有相同结构的基于GRU RNN模型的Decoder和辅助Decoder进行标签生成。在PDTB2.0数据集下,测试集上的macro F1值最好能够达到65.63%。

1 实验简介及实验环境

1.1 隐式篇章关系判别

隐式篇章关系判别(implicit discourse relation recognition, IDRR)即是判断两个文本段之间隐含存在的关系,属于自然语言理解任务中篇章分析领域的重要理论,对于提升机器阅读能力具有重要作用[1]。

根据PDTB语料库中提出的术语,一个IDRR任务中通常都会涉及以下的几个部分:

- Argument: 语篇中的一个文本片段,包含至少一个谓词,用于陈述一个问题、事件或观点
- Sense: 一个语篇关系的类型,如Comparison、Expansion、Temporal、Contingency等
- Connective: 表达两个Arg之间关系的连接词,如”but”等

大多数情况下,两个Argument之间并不存在显示的连接词,从而将这些不存在显式连接词的论元间关系称为隐式篇章关系。

1.2 PDTB2.0 语料库

PDTB语料库被认为是篇章关系识别任务中最大的语料库,标注了来自《华尔街日报》的超过100万单词[2]。根据PDTB2.0语料库的标注手册[3]显示,其标注分为了三层的level: *Class* → *Type* → *Subtype*,如图一所示。其中第一层共计有4中Class,细分第二层共有16类Type。在本实验中,主要关注的是模型在第一层:4类Class上的性能表现。

*Email: xiaoliangshou.bit@gmail.com

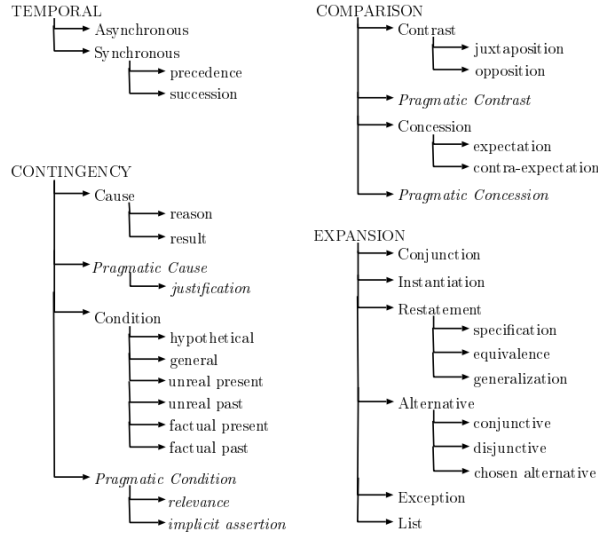


图 1: PDTB层次结构

1.3 评测指标——MacroF1值简介

F1值，也称F-measure，是十分常用的用于评估分类模型性能的一种度量方式。

在较早的机器学习任务中，为评判二分类模型优劣，常用一个“混淆矩阵”(Confusion Matrix)来表示分类的结果，如下图所示：

从中可以定义出两个概念——查准率P与查全率R，各自的公示表示为：

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}$$

F1值便是基于查准率P和查全率R而引入的一种更为全面、科学的度量方式，数学表达方式写为：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{N + TP - TN}$$

从原理上讲，F1值是基于P和R的调和平均，即

可以表示为：

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{P} + \frac{1}{R} \right)$$

在多分类的情况下，则需要采用取平均的方式来计算F1值，由此也产生了多种平均F1值得计算方式，最常见的有三类：宏观F1值(macro F1)、加权F1值(weighted F1)、微观F1值(micro F1)值。宏观平均的方法是三者当中最直接的方式——假设共有N个类别，每个类别的F1值分别为 $F1_0, F1_1, F1_2, \dots, F1_{N-1}$ ，则求出所有F1值得算数平均即为宏观平均，即 $macroF1 = \frac{1}{N} \sum_{i=0}^{N-1} F1_i$ 。

不同于前者认为每个类别等权分配，加权F1值则是考虑了每个类别的支持度(support)——即该类在数据集中实际出现的次数(观测值)。

1.4 预训练RoBERTa模型

RoBERTa[4]是对预训练模型BERT的延伸和改进。BERT的基本结构是Transformer，将两个片段(token序列)， x_1, \dots, x_N 和 y_1, \dots, y_M 的连接作为输入，每个片段通常由一个以上的自然语言句子组成。在将这样的片段输入给BERT之前，还需要用一些特殊的符号来分隔它们，即 $[cls], x_1, \dots, x_N, [SEP], y_1, \dots, y_M, [EOS]$ ，其中 $M + N < T$ ，T用于控制训练过程中最大的序列长度。在BERT预训练过程中，追求的目标有两个：遮掩语言模型(masked language model, MLM)和下一个句子的预测(next sentence prediction, NSP)。

BERT模型采用Adam作为优化器，其参数配置为： $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e - 6, L2 = 0.01$ ，学习率在前10000步迭代中先加热到峰

值 $1e-1$ ，然后线性衰减，所有层的 $dropout$ 设置为0.1，使用GELU激活函数，最小batch设置为 $B = 256, T = 512tokens$ 。

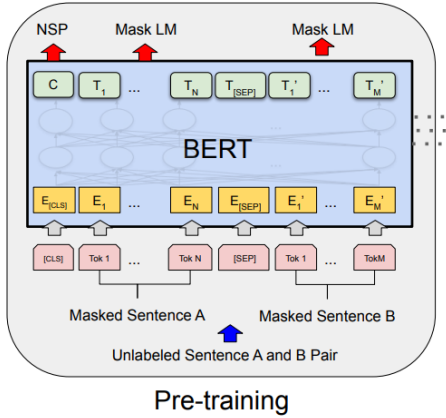


图 2: BERT预训练模型

RoBERTa模型对BERT作出的改进主要体现在三方面：(1) 改进了优化函数，采用更大的batch， β_2 下调为0.98；(2) 训练策略上，采用动态掩码来训练模型，每次向模型提供输入时动态生成mask (3) 数据层面上，使用了更大的数据集，并且使用BPE(Byte-Pair Encoding)来处理文本数据。¹

1.5 实验环境

本次实验借助了华为弹性云服务器进行模型训练，服务器的硬件配置如下图所示。

2 模型结构与相关原理

实验所用的模型为Changxing Wu等于2021年

¹RoBERTa预训练模型下载链接：<https://huggingface.co/roberta-base/tree/main>

```
root@ecs-xls1:~# nvidia-smi
Tue Dec 28 18:38:29 2022
```

NVIDIA-SMI 470.161.03 Driver Version: 470.161.03 CUDA Version: 11.4									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr.	ECC	GPU	Util	Compute M.
Id	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	Id	Temp	Perf
0	tesla T4	Off	00000000:00:00:00	Off	0%	Default	0	tesla T4	Off
N/A	69C	P0	30W / 70W	0M1B / 15109M1B	0%	Default	N/A	69C	P0
1	tesla T4	Off	00000000:00:00:00	Off	0%	Default	1	tesla T4	Off
N/A	59C	P0	31W / 70W	0M1B / 15109M1B	0%	Default	N/A	59C	P0

```
Processes:
GPU  GI  CI  PID  Type  Process name  GPU Memory
ID  ID
No running processes found
root@ecs-xls1:~# ls
```

图 3: 华为云服务器硬件配置

提出的LDSGM模型[5]：标签依赖感知序列生成模型(Label Dependence-aware Sequence Generation Model)²。

通过以上对PDTB语料库的简介可知，PDTB语料库的标签分为了三级，模型的基本思想是，给出M层由标签 $C = (C^1, \dots, C^m, \dots, C^M)$ 定义的分层levels，其中 C^m 代表第m层上的标签的集合。将一个实例 $x = (arg1, arg2)$ 作为输入，模型产生一组标签序列 $y = y_1, \dots, y_m, \dots, y_M$ ，其中 $y_m \in C_m$ 。总体上看，LDSGM模型由一个Label Attentive Encoder和一个Label Sequence Decoder组成，模型的结构如图三所示。

2.1 Encoder

在Encoder部分，包括几个堆叠的Transformer层、一个图卷积神经网络(GCN)和特定level的Label Attention机制。各部分的具体功能如下：

1. 模型使用Transformer层学习输入实例的局部和全局表示。对于给定的实例 $x = (arg1, arg2)$ ，首先堆叠K个Transformer层来

²原文源代码发布于<https://github.com/nlpersECJTU/LDSGM.git>

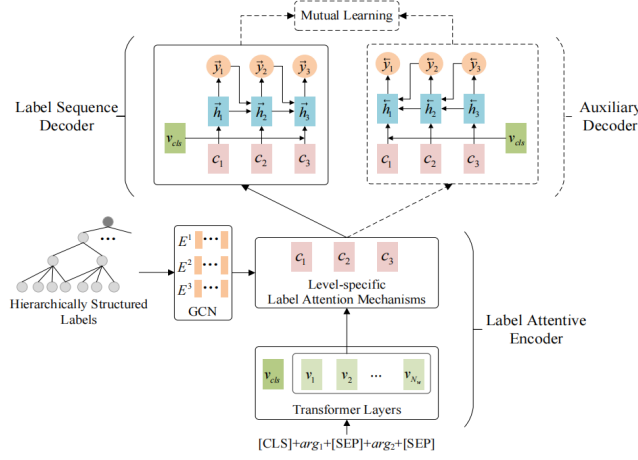


图 4: LDSGM模型结构

学习word-level的语义表示:

$$v_{cls}, v_1, \dots, v_{N_w} = \text{Transformer}(arg_1, arg_2)$$

其中实例需要以如下的特殊格式表述: $[CLS] + arg_1 + [SEP] + arg_2 + [SEP]$, v_{cls} 代表[CLS], $\{v_i\}_{i=1}^{N_w}$ 表示其他的token, Transformer的主要部分是多头Attention。在后续的预测任务中, v_{cls} 被认为是实例的全局表示, 而 $\{v_i\}_{i=1}^{N_w}$ 则被认为是局部表示, 从中提取到特定的上下文关系, 用于不同level的预测。

2. 使用GCN通过集成层次结构化的标签之间的依赖关系, 来获得更好的标签嵌入(label embedding), 最后使用标签注意机制从局部表示中提取特定于级别的上下文。
3. 完成上述工作之后, 模型将学习到的全局表示和特定于级别的上下文作为Decoder的输入, 用于生成标签序列。

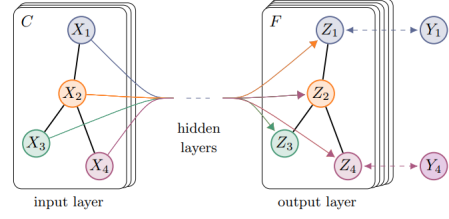


图 5: GCN结构

2.2 Decoder

Decoder是一个基于RNN的解码器, 自顶向下依次生成预测标签, 如此可以借助更容易预测的更高层次的标签(例如PDTB数据集中分为四类的第一层标签)。构建Decoder使用的是广泛用于文本生成和短长度标签序列的门循环单元GRU。据R-Net的提出者给出的观点[6], GRU与LSTM在实验效果上能达到相似的结果, 但是GRU相较于LSTM更容易计算。在计算资源并不富裕的情况下, 选择GRU更为合适。

2.3 互学习增强训练

互学习增强训练(Mutual Learning Enhanced Training)模块是紧接着Decoder的最后一个模块。Decoder以自顶向下的方式生成一个标签序列, 然而其只能利用来自预测出的更high-level的依赖关系, 而来自lower-level的依赖关系却不被利用。本部分采用自底向上的标签依赖关系来改进模型, 可作为Decoder的这一点瑕疵的补充——跟据作者提供的实验结果来看, 不加入互学习增强模块时模型已经能够取得很好的效果。

具体来说, 本部分是一个辅助Decoder, 与Decoder具有相同的结构, 但是其顺序产生的标签序列为 $y = y_M, \dots, y_m, \dots, y_1$, 与Decoder相

反。Decoder和辅助Decoder能够从两个不同的方向捕获标签依赖关系，因此认为二者是互补的，从而达到互学习增强的目的。

训练过程如算法1所示。为实现Decoder和辅助Decoder训练过程中的相互促进增强，除了交叉熵损失外，作者还额外引入了两个额外的损失，用以减少两个Decoder之间的分歧。具体来说，假设 $\theta_e, \theta_d, \theta_{ad}$ 分别代表Encoder、Decoder、辅助Decoder的参数集，在训练数据集 D 上定义如下目标函数来更新这些参数：

$$L(D; \theta_e, \theta_d) = \sum_{(x,y) \in D} \sum_{m=1}^M \{ -\mathbb{E}_{y_m} [\log \vec{y}_m] + \lambda * KL(\vec{y}_m || \hat{\vec{y}}_m) \} \quad (1)$$

$$L(D; \theta_{ad}) = \sum_{(x,y) \in D} \sum_{m=1}^M \{ -\mathbb{E}_{y_m} [\log \hat{\vec{y}}_m] + \lambda * KL(\vec{y}_m || \hat{\vec{y}}_m) \} \quad (2)$$

其中 $\mathbb{E}_{y_m} (*)$ 代表 y_m 的期望， $KL(*)||*$ 是KL散度公式， λ 用于控制不同损失项的影响系数。

3 实验内容及结果

在LDSGM模型的原文中，由于只有部分PDTB实例标注了第三级的标签(Subtype)，并且16个二级标签(Type)中有5个去掉了很少的训练实例，缺少验证实例和测试实例，因此遵循IDRR研究中的惯例，只将第一二级标签考虑在内，考虑11类Type分类，并将插入的连接词作为第三级标签。本次实验只需要评估模型在一级(Class)上的表现，由此可带来一定的简化。参数设置如表1所示。

Algorithm 1 算法训练过程

Input:

Training set D , Test set D'

```

1: repeat
2:   repeat
3:     load a batch size of instances  $B \in D$ 
4:     Generate predicted label distributions
        $\vec{y}_1, \dots, \vec{y}_m, \dots, \vec{y}_M$  using the decoder for
       each instance in  $B$ 
5:     Generate predicted label distributions
        $\vec{y}_M, \dots, \vec{y}_m, \dots, \vec{y}_1$  using the auxiliary de-
       coder for each instance in  $B$ 
6:     Update  $\theta_e, \theta_d$  by minimizing  $L(B; \theta_e, \theta_d)$ 
7:     Update  $\theta_{ad}$  by minimizing  $L(B; \theta_{ad})$ 
8:     Save the best model according to the av-
       erage performance at all levels on  $D'$ 
9:   until no more batches
10: until convergence

```

在修改前训练总计15次epoch后的运行截图6如图所示。同时，更改后的运行结果如表2所示其中的f1值指macro F1。从中可以看出，训练完成之后，测试集上的最好的F1值可达到65.63%。

更为详细的输出日志可参见”Dataset/log/”目录下的输出日志。

训练得到的权重模型可从google硬盘下载，链接如下: Google Drive Adress。

```

train time usage: 570.49350408955547
test time usage: 5.4077847089594677
TOP: test Loss: 4.6, Test Acc: 72.66%, Test F1: 65.75%
SEC: test Loss: 4.6, Test Acc: 59.18%, Test F1: 58.83%
CONN: test Loss: 4.6, Test Acc: 40.15%, Test F1: 11.79%

precision recall f1-score support
Temporal 0.5608 0.4783 0.5197 69
Contingency 0.7500 0.5724 0.6488 275
Comparison 0.6242 0.7103 0.6645 145
Expansion 0.7618 0.8572 0.7973 559

accuracy 0.7266 1046
macro avg 0.6495 0.6575 1046
weighted avg 0.7265 0.7218 1046

precision recall f1-score support
Temporal.Asynchronous 0.5862 0.6182 0.6018 55
Temporal.Synchronous 0.0000 0.0000 0.0000 14
Contingency.Cause 0.7004 0.5955 0.6424 268
Contingency.Pragmatic cause 0.0000 0.0000 0.0000 7
Comparison.Contrast 0.5318 0.7244 0.6155 127
Comparison.Concession 0.0000 0.0000 0.0000 17
Expansion.Conjunction 0.5288 0.5622 0.5446 201
Expansion.Instantiation 0.7658 0.7283 0.7424 118
Expansion.Restatement 0.5545 0.5877 0.5598 211
Expansion.Alternative 0.3529 0.5867 0.4615 9
Expansion.List 0.1429 0.0855 0.1053 12

accuracy 0.5918 1059
macro avg 0.3768 0.4142 0.3985 1059
weighted avg 0.5791 0.5910 0.5811 1059

dev_best_acc_top: 70.15%, dev_best_f1_top: 62.05%,
dev_best_acc_sec: 59.05%, dev_best_f1_sec: 55.97%,
dev_best_acc_conn: 32.12%, dev_best_f1_conn: 9.13%
root@ecs-xls:~/xls/LDSGM#

```

图 6: 修改前运行截图

表 1: 实验参数设置

参数名	参数值	参数意义
n_top	4	Class的数量
pad_size	100	最大句子长度
num_epochs	15	迭代次数
learning_rate	1e-5	学习率
bert	Robert	Transformer预训练模型
batch_size	32	batch的大小
hidden_size	768	隐藏层的大小
x_dim	768	输入数据的维度
num_gcn_layer	2	GCN的层数
label_embedding	100	标签嵌入的维度
attn_hidden_size	768	Attention隐藏层的大小

表 2: 训练结果

epoch	3	6	9	12	15
train acc	57.14%	85.71%	100.00%	100.00%	85.71%
test f1	60.59%	62.91%	63.74%	65.18%	65.63%
val f1	58.00%	58.25%	60.03%	60.65%	62.24%

参考文献

- [1] W. Xiang and B. Wang, “A survey of implicit discourse relation recognition,” *ACM Comput. Surv.*, dec 2022, just Accepted. [Online]. Available: <https://doi.org/10.1145/3574134.1.1>
- [2] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber, “The penn discourse treebank 2.0.” in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, 2008. 1.2
- [3] P. R. Group *et al.*, “The penn discourse treebank 2.0 annotation manual,” *December*, vol. 17, pp. 26–37, 2007. 1.2
- [4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019. 1.4
- [5] C. Wu, L. Cao, Y. Ge, Y. Liu, M. Zhang, and J. Su, “A label dependence-aware sequence

generation model for multi-level implicit discourse relation recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 486–11 494. 2

- [6] Y. Wang, H. Xie, Z. Zha, Y. Tian, Z. Fu, and Y. Zhang, “R-net: A relationship network for efficient and accurate scene text detection,” *IEEE Transactions on Multimedia*, vol. 23, pp. 1316–1329, 2021. 2.2