S+PAGE: A Speaker and Position-Aware Graph Neural Network Model for Emotion Recognition in Conversation（CoRR，2021Dec）

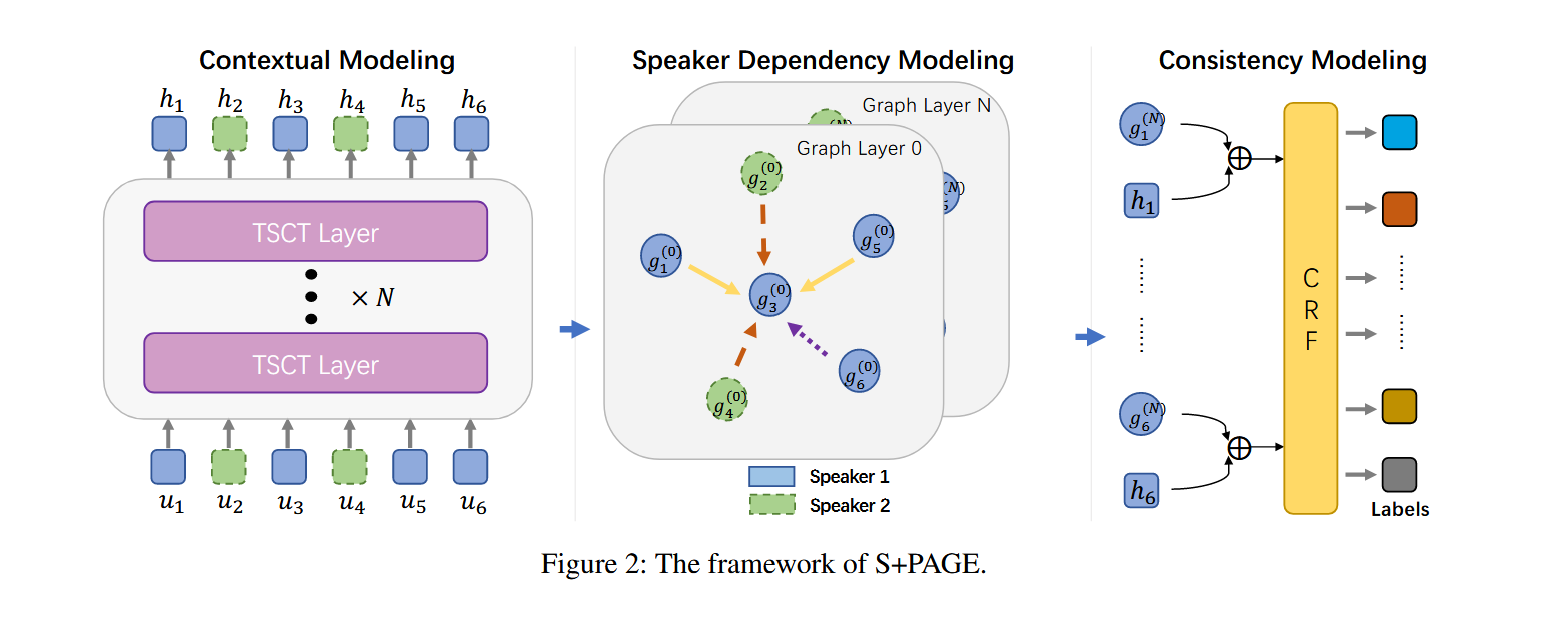
（这个是专门用于提取情绪的，Neutral Utterances are Also Causes: Enhancing Conversational Causal Emotion Entailment with Social Commonsense Knowledge 又是依据情绪提取原因的，感觉可以合并一下，而且都是用上图神经网络的，下文都是两者不同方法的比较）

Motivation：

现有的 ERC 方法大多分别对自我和对话者之间的上下文进行建模（感觉除了图类的，很少注意说话人这个信息），这就带来了一个主要问题，即它们之间缺乏足够的交互。从2019年Poria论文发现对话上下文在ERC任务中，在2020年前的研究多使用RNN处理小范围的对话上下文，因为话语大多是连贯的，其效果在大多情况下都有很好的性能。但正如我们所说，RNN由于其过多的参数和计算次数，在远距离的上下文的信息收集，大规模的多方对话上就力不从心了。

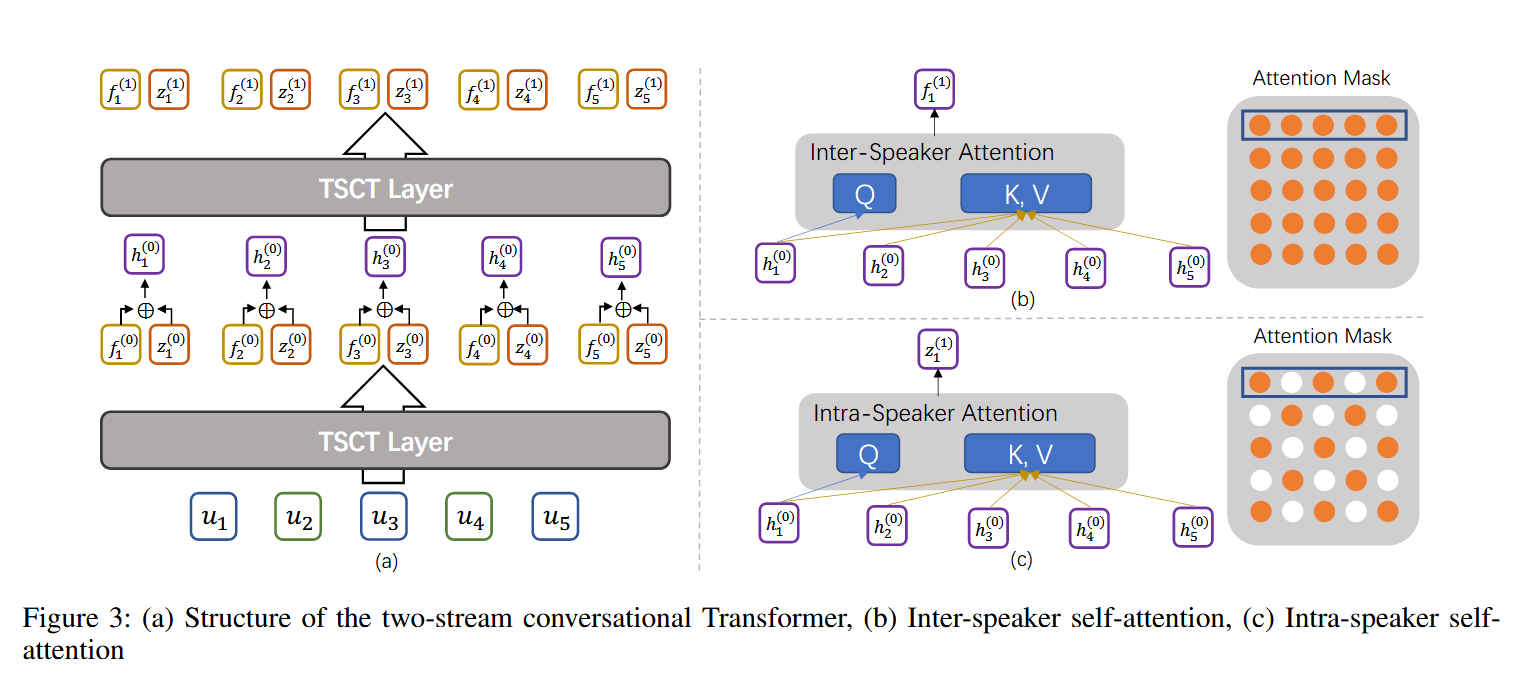
为了解决上述问题提出了使用GNN，解决RNN不能结合多方对话的缺点。但同时使用GNN的时候也要关注自我说话人的语境影响，也要关注其他说话人的影响，但是由于一般的GNN算法中，只单单区分了边的有无，而不保留边的距离信息。或者如分层对比学习中提到的，不要过于局限于距离，而是直接分析不同有关信息的权重分布。所以本文方法就是基于探讨添加相对距离会不会提高模型性能。

我们的模型包含三个阶段，以充分考虑自我和说话人间的语境。具体来说，给定同一对话中的一系列话语，我们首先利用具有细心屏蔽机制的双流对话转换器 （TSCT） 来获得同一说话人和整个对话的上下文特征。然后，在说话人身份和话语顺序的指导下，我们构建了一个说话人和位置感知对话图。位置感知 GNN 模型 （PAG） 是一种增强的关系图卷积网络 （R-GCN），用于优化具有自依赖性和说话人间依赖性的上下文特征。特别是，我们在聚合算法中引入了关系相对位置编码，以使 PAG 能够捕获细粒度的时间信息。最后，情绪标签的全局传输由条件随机场 （CRF） 层建模，该层具有来自 TSCT 和 PAG 的特征。

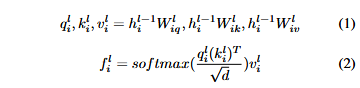


主要分为三个部分第一个部分，说白了就是获取每个句子的表示，是一个n层的Transformer，获得粗略的上下文特征以及说话人信息的提示，也就是隐藏层表示。【而在对比论文中是RoBERTa】第二部分，提出了一种图神经网络，以使用说话人依赖性和由关系相对位置特征指导的时间信息来微调特征。（说的直白点就是1、一个节点应该与那些节点连接。2、每个节点的权重是多少。）第三部分，我们采用条件随机场来模拟全局情绪一致性的背景，即情绪转移。自我和说话人间情境之间的互动涉及前两个阶段。

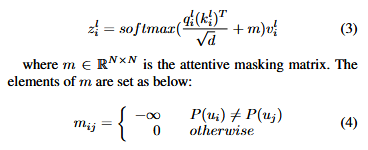
话语编码部分：第一步是使用Glove向量得到词嵌入，然后使用一个前馈神经网络，（这里特定使用没有上下文提取的网络），得到句子的表示。

 第二步是Two-Stream Conversational Transformer，双流对话转换器，结构如图三（a）所示，它的输入是上一轮的得到每个句子的，不带上下文信息的表示U[h]。之后经过多头自注意力机制，分别处理说话人间信息和说话人自我信息。

说话人之间信息的自注意力机制和Transformer的实现一样，上下文窗口和其他论文一样，只在这句话的所在对话中，（比如说在一组对话中，总共5句话，那么每个人的自注意力机制所包含的节点个数就是句子个数）。计算公式如下：一共有L层



说话人内部自我注意力机制：仅计算对话中同一说话人话语的注意力来模拟说话人特定的上下文信息，其上下文窗口同样也是一组对话中所包含的句子个数，通过一个二维识别矩阵M记录每一句话的说话人与其他那些句子是一样的，然后只对自已说的话进行自注意力计算相同的时候不足修改，不同的时候赋值负无穷，让注意力计算的结果为0.



每一层的Transformer都需要将每个节点得到的两种注意力信息进行合并，最后将经过多层操作得到的结果进行一个维度的修正，（因为随着维度的合并，维度会翻番，但是每一层之间要不要修正没有体现出来，）最后经过层归一化得到最终的输出。

位置感知图神经网络：  
 通过引入相对位置编码来捕获说话人的依赖性和事件信息，图架构：一个图只有一组对话的信息，分别有代表句子特征向量的节点，代表说话人身份信息和相对距离定义的两种信息的边，以及边的权重信息。

图神经网络也包含多层，每个图节点的初始值都是由第一部分的话语表示得到，每个节点都有对应的上下文窗口，上文窗口为p，下文窗口为f，不同节点之间以边相连。边的信息包含距离大小和说话人信息，边的权重由注意力机制计算得到，其输入是在上下文窗口内的节点表示，加上相对位置编码项，计算公式如下所示：



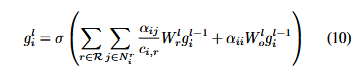


Ｗ矩阵维度为［相对位置编码维度F’，句子表示维度F］，是可学习矩阵，然后加上话语i和话语j之间的位置编码β[i，j]，



其中o()是一个映射函数，将一句话表示为他在对话（没细说的话，应该只是在i这个上下文吧）中的绝对位置（是形成一个01向量还是啥，没有说明），再通过Ep矩阵，维度[上下文窗口大小,相对位置编码维度]。

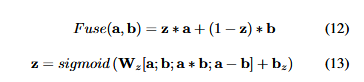
然后将话语i和话语j的结果进行拼接，a是一个参数化的权重矩阵，维度[1,2F’]，最后得到每个节点的值，经过softmax算概率。



网络中的节点更新公式如上，遍历所有边的关系所代表的边，（边关系包括了是否为同一说话人，话语j相对于话语i的距离）c[i,r]表示归一化常数，对于每一种对话，每一种边关系有一种值，大小就是话语i和话语j之间的距离绝对值。W是可学习矩阵，维度是[F,F]，a[i,j]和a[i,i]就是上文计算的边权重大小，最外层套着激活函数，共计算L层。

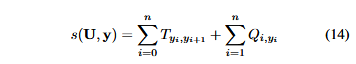


对于L >= 1层的节点表示，还可以融合前一轮的节点表示。

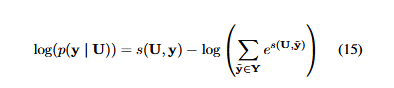


a\*b和a-b应该是数值上的相乘，然后将他们全部相加还是合并（没有具体说，我感觉就是直接数值对应加起来吧，bz也没说是什么。。应该是一个数字吧）然后得到融合比例。

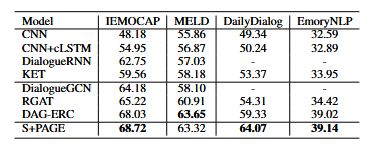
条件随机场 为了模拟标签一致性，即对话中的情感转移，采用线性链条件随机场来产生每个话语的最终情感标签。U表示原始的句子，y表示每个句子对应的情感标签，句子个数为N，标签种类为K。



这个序列的得分计算公式如上：T是一个可学习的矩阵，维度[K，K]，Q是一个可学习的矩阵，维度[N,K]，总之结果是一个数字。

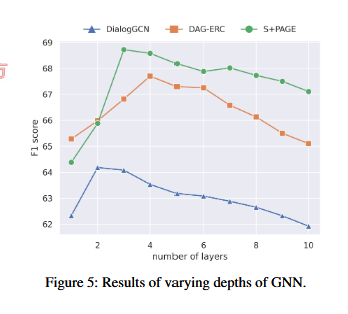


最后模型需要最大化上述式子计算的值，也就是最小化损失函数。其中的U换成了前两个模型的最终输出。

实验结果：  


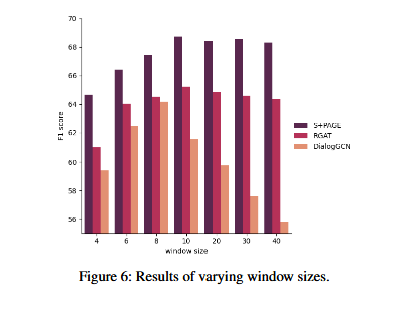
可以发现由于1） 具有自我和说话人间依赖性的上下文建模，2） GNN 中存在相对位置编码，3） 全局情绪转移的一致性建模，使得模型的判断精度得到提升，但是在MELD，EmoryNLP这些依据老友记记录的对话，由于涉及大量常识性知识，表现反而下降。

在消融实验中发现，PAG（位置图神经网络）对模型的提升效果最大，说明将话语与上下文良好结合的话，对情绪的识别精度会得到较大提升。并且测试了不同深度的PAG的影响：



可以发现，网络层数过大的话，可能会出现过拟合的现象，所以需要设置合适的层数。并且不同的网络所需要的初始句子表示也是不同的，本文的句子表示使用两种注意力机制混合得到的Transformer，而换成RoBERTa得到的句子表示，性能也会下降。同样的，不同的上下文窗口也会有影响，当上下文窗口再合适范围内，增加大小会包括更多有用信息，而过大之后，无用的上下文信息就会干扰模型了。

所以说，合适的范围，以及合适的表示也是重要的。



ECPEC: Emotion-Cause Pair Extraction in Conversations

对话中的情绪-原因对提取（TAC，2022，ccf-b）

Motivation：

近年来，对话情感分析 （CSA） 和情绪原因对提取 （ECPE） 任务越来越受到关注。前者旨在预测对话中说话人的情绪状态，后者是关于提取文档中的情绪原因子句。然而，CSA 的一个缺点是它无法模拟来自不同说话者的情绪和中性话语之间的因果推理。在这项工作中，我们提出了一项新任务：对话中的情绪-原因对提取 （ECPEC），旨在提取对话中的情绪话语对和在不预先知道情绪类别的前提下找到相应的原因话语。并且设计了新的数据集ConvECPE，以及针对这个数据集的两步框架。

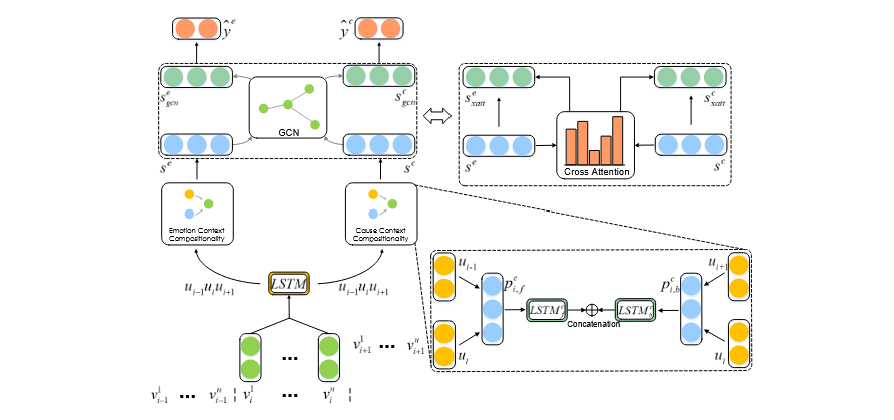
在对话中的ECPE任务相较于基于文档的ECPE任务来说，非语法性、不连续性、上下文依赖性和交互性等独特属性使对话数据更难分析，比如说情绪-原因对在一组对话中可能有很多，或者说一句话既是情绪又是原因（比如“我感觉不舒服。我失业了。”句子给出了情绪，也指明了原因。），或者说一句话的组成可能只是语气词，或者是短语。ConvECPE数据集的每个非中性话语至少有 1 个 cause 话语，最多有 3 个 cause 话语。我们不标记中性话语，因为大多数中性话语之间的情绪交互相对较弱。

情绪话语与原因话语的距离经过测算处于[-42,39]的范围内，最大值多分布在[-2,6]之间，说明情绪多存在与原因之后，但也有不少是情绪出现在原因之前的情况。所以较小的上下文窗口也足以提高模型精度了。而对于有多个原因句子导致的情绪子句来看，有大约80%左右的原因句子之间距离不超过3.也就是说可以在合适的范围内搜索原因句子，而不失准确性，所以提出了分块协助检测。

模型介绍：

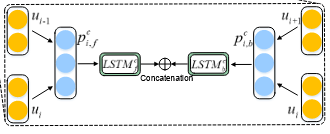
第一步：联合情绪原因检测，只需要检测给出的句子是不是情绪，以及是不是对话中任何句子的原因句子。为此定义了两种多任务学习网络，共同提取对话中的情感和引起的原因句子。第二步：EC 块对提取和 EC 对提取，简单的使用笛卡尔积一一判断所有取值情况会导致很多不必要的判断，所以依据对数据的统计研究成果。情绪句子相关的原因句子大多集中在整段对话的一小部分，那么我们直接对这段对话中的原因句子集合进行拆分成多份原因句子集合，此时情绪句子会与原因句子集合进行配对，只需要保证配对的原因句子在这个集合中即可。之后只需要遍历这个小集合内的原因句子与情绪句子配对就 可以了。

模型1 的整体架构如下：

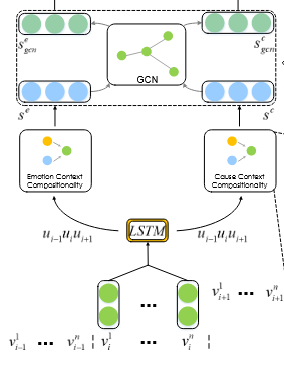


提出了两种联合情绪和原因话语检测模型。Joint-GCN 如左侧所示，它使用图卷积网络作为软信息共享方法，以增强情绪和原因检测任务。Joint-Xatt 模型的结构与 Joint-GCN 相似，不同之处在于 Joint-Xatt 模型采用交叉注意力作为软信息共享方法，以提高两项任务的性能。以 Joint-GCN 为例，它分别有 emotion 和 cause context 组合性模块，分别用于 emotion 和 cause 检测任务。这两个模块的结构相同，上图只是原因句子检测部分。

模型1：联合情绪原因检测，（这里是以一轮对话作为一组输入，不是全部的对话）先使用Glove词向量转换，然后使用LSTM生成句子的表示。然后将话语i的表示分别与它的相邻话语对u[i-1],u[i+1]作为两个输入，将他们输入一个线性层，它将输出4个结果，获得用于情绪检测任务的上下文感知向量 p[e,i,f]和 p[e, i，b]以及用于原因检测任务的 p[c,i，f] 和 p[c, i，b]。然后将情绪和原因的上下文感知向量，以输入进前向和后向的LSTM模块中，然后将两个的结果融合，（也就是一个另类的BiLSTM方法），图示如下：左边前向，右边后向。只以计算原因句子表示为例子，计算情绪句子同理。

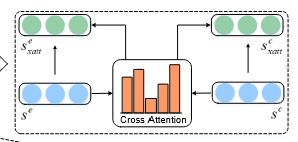


合并后得到的输出记为S[e,i]和S[c,i]，然后传递到下一个部分图卷积网络中

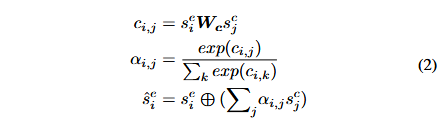


研究发现，情绪信息的引入可以助于检测原因，反之亦然。或者软参数共享也可以提高多任务学习模型的性能，为此提出了两种具有不同软信息共享机制的多任务学习变体，以增强检测性能。

第一部分是注意力机制部分，Se表示经过前面的LSTM组合句子表示得到的对应与情绪检测任务的句子表示集合，Sc同理是原因句子表示集合。

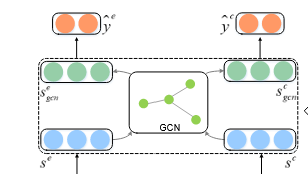


而注意力机制会为每一个情绪句子表示附加额外信息，具体计算公式如下：（k好像是指一轮对话中的句子个数）W是可学习参数，维度[Du,Du]，通过将两个情绪，原因句子表示进行矩阵运算得到权重，a[i,j]平均所有范围内的计算记过，最终得到的句子表示为[Du,2]。（范围j好像没有细说。），反正将多个与原因句子表示进行拼接，得到附加的情绪句子表示信息S[e,xatt]。原因句子同理S[c,xatt],然后将得到的附加信息集合放入全连接层进行分类。



损失函数同样使用交叉熵损失。

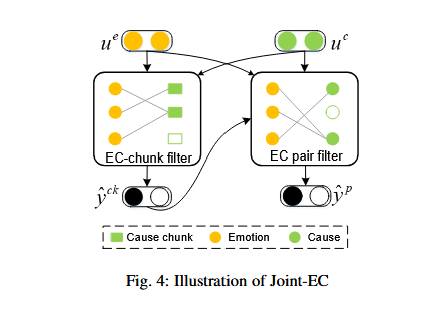
第二部分是图卷积网络部分：



这里的输入依然是从LSTM部分得到的句子的情绪表示和原因表示Se和Sc。图中的每一个节点代表了这两个集合中的一个向量，而边只会连接在来自集合Se和Sc的向量且有依赖关系的表示的组合之间，（这里的依赖关系是指的两者是不是一组情绪原因对，有关系才会连接边吧，它论文又不说清楚。）然后将这个图信息输入GCN图卷积网络中，（具体实现细节省略，可以参考L. Yao, C. Mao, and Y. Luo, “Graph convolutional networks for text classification,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33(01), 2019, pp. 7370–7377.和D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, “Dialoguegcn: A graph convolutional neural network for emotion recognition in conversation,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.）总之，最后的结果又会得到每个句子的情绪表示和原因表示S[e,gen]和S[c,gen]同样放入一个全连接层进行分类，损失函数由判断情绪句子和原因句子的交叉熵函数，按照一个超参数调整比例后得到。也就是说信息融合可以选择用注意力机制，也可以用图卷积网络GCN。

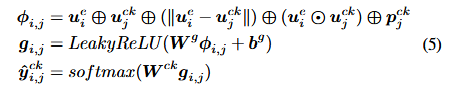
图卷积网络GCN简单描述：  
输入是一个图邻接矩阵，每个节点需要是一个特征向量。内部的实现就是在节点的相邻节点之间传递信息来更新节点的特征表示，通过对邻居节点的特征进行加权求和来更新当前节点的特征，如果需要还可以堆叠多层GCN，最后得到每个节点最后的输出。

模型2：情绪-原因块和情绪-原因对的提取

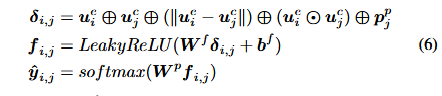
经过模型1的判断后，我们得到了一组对话中的情绪句子集合E，和原因句子集合C。  


左边的模块为EC块的过滤器，右边模块为EC对的过滤器。EC块的输入只有经过模型1后找到的情绪，原因句子集合的话语级表示，就是只经过了Glove词嵌入，和LSTM得到的结果。而EC对的输入在此基础上增加了EC块过滤器的输出。

EC块过滤器的计算流程和公式如下：

  
 ⊙代表元素乘法，也就是维度的每个元素直接相乘。U[ck,j]表示第j个原因块的嵌入，而原因块的嵌入是通过拆分原因句子的集合得到的，并应用了注意力机制整合这些嵌入得到一个和情绪句子表示相同的维度。P[ck,j]是这个EC块的位置嵌入表示。然后将5种不同的表示进行拼接，经过一个全连接层得到情绪句子i对应与每个原因对的表示g[i,j]然后经过维度变化，应该是变到维度 = 一组对话种的原因块个数，这样softmax的结果，就是这个情绪句子i对应哪一个原因块了。

然后在情绪句子i中的原因块中抽取它所能配对的EC对，EC过滤器的流程公式如下：

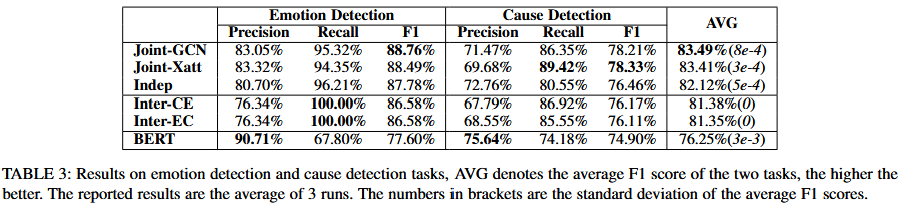


和EC块中的流程大致一样。将EC块表示，换成具体的原因句子表示即可。此外还可以在计算的时候，可以使用教师机制，保留部分真值。或是在Joint-EC(联合情绪-原因识别)使用窗口限制，将（u[c,i] - u[c,j]），i-j ∈ [-window,window]作为输入（这个窗口大小是改变了Joint-EC在每一块内的判断输入吗？？）。因为两者的输出结果仍是一个数，所以同样使用交叉熵作为损失函数。

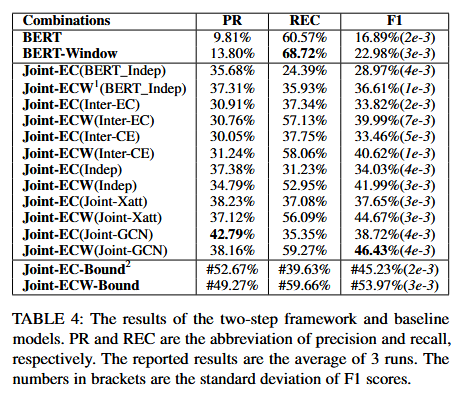
实验结果：

第一次实验评估了模型1在预测情绪和原因句子上的性能，分别测试了实验联合图卷积网络，联合注意力机制，分别预测三种方法的性能，联合得到的性能会比分别更加好，而图卷积网络比注意力机制好一丢丢。整体比最初提出的EC判断方法更优。

而且本文也注意到了Inter-CE这些方法是将预测结果直接作为另一个任务的输入特征的一部分，而MECPE则是直接共享了输入特征，但是本文提出的方法则是将软信息共享。总之，在情绪，原因识别的时候，最好能避免分出多个模型，而是尽量整体架构，UniMEEC就是一个完全整体架构出来的，之前的模型，多多少少还是分成两步。

 此外，BERT 倾向于尽可能精确地检测情绪或引起话语，而不是尽可能多地检测情绪或引起话语，Inter-CE则是可能由于数据集中有很多情绪化句子，误导其将大多句子识别为情绪句子，而不是原因句子。并且发现，情绪句子判断精度远高于原因句子判断精度，因为情绪句子有专门的情绪词提供足够信息，而原因句子要想获取足够信息就必须从情绪句子中来，正如Hierarchical Shared Encoder with Task-specific Transformer Layer Selection for Emotion-Cause Pair Extraction 所述。那这个方法就是解决了这个问题。但是这个文章的情绪-原因对预测就是硬找了，反而可以用这个方法，可以比较一下用原因块限制，和不限制的区别。

接下来的就是将情绪，原因判断模块，融合进不同的情绪句子，原因句子识别模型中，结果如下图所示：



可以发现，应用窗口限制，会让模型得到不错的性能提升，因为整体判断个数下降了，而真正的EC对个数不变。综合两个实验结果也可以发现，情绪句子和原因句子的提取结果将会决定EC对判断的精度，因为正确的情绪，原因句子越多，预测正确的概率就越高。所以尽可能提高情绪句子，和原因句子识别精度是重要一环，可考虑的方法就是加入先验知识，这就是社会常识知识融入的效果了。而提出先找EC块，再找EC对的方法同样有类似效果。

并且块大小选择合适大小对模型提升最大，过小的话包含的信息有限，而过大会包括过多不相干信息干扰模型。但不足的是，块的限制能一定程度上提取更多候选对，但也无法识别长程EC对。而且我们研究的是对话中的情绪-原因对识别，前面的情绪句子不可能是有后面的原因句子引起，但是本模型不能避免这个问题。

结合起来就是，先识别情绪句子,可以用本文的方法或是使用第4篇论文Hierarchical Shared Encoder的方法，然后拆为两种流派，第一种是再识别原因句子有哪些，同样使用第4篇论文Hierarchical Shared Encoder方法。再将情绪句子组合其所在对话中的文本作为输入，经过Transformer后与对应情绪一起放入网络，或是和本篇一样联合学习。然后用本篇的方法，或许可以使用MECPE中提到的融合社会常识知识。第二种就是跳过识别原因句子，直接通过情绪句子识别情绪-原因对，就是第五篇论文Neutral Utterances are Also Causes的方法了。所以接下来就是看这几篇的代码。

不过现在也有大模型来做的，比如UniMEEC，两种都看看。下一篇就看看这个用大模型来prompt-tuning有关的吧。

EmoPrompt-ECPE: Emotion knowledge-aware Prompt-tuning for Emotion-Cause Pair Extraction（COLING，ccf-b，2024）

Motivation：

情绪-原因对提取 （ECPE） 的主要重点是从未注释的文档中提取所有潜在的情绪子句和相应的原因子句。现有方法在微调和提示范式的帮助下取得了有希望的结果，但它们存在三个缺点。首先，大多数方法无法区分属于不同类型情绪的情绪-原因对，这限制了现有方法的适用性。其次，现有的提示方法利用一对一的映射关系来实现标签词到类别的映射，这给结果带来了相当大的偏差。第三，现有方法实现了由显式语义理解或基本提示模板支持的原因提取任务，忽略了 原因子句本身包含的隐式信息。

解决方法：  
 提出了一种情绪知识感知的情绪-原因对提取提示调优 （EmoPrompt-ECPE） 方法，该方法将情绪类别的知识整合到 ECPE 任务中，并挖掘原因子句的隐性知识。具体来说，我们将原因子句和 情绪句子的潜在知识注入到 prompt 模板中。此外，我们将标签词的多对一映射的情感标签扩展到具有外部情感词基的类别。此外，我们利用标签词库的余弦相似度过滤来减少知识引入带来的噪声。

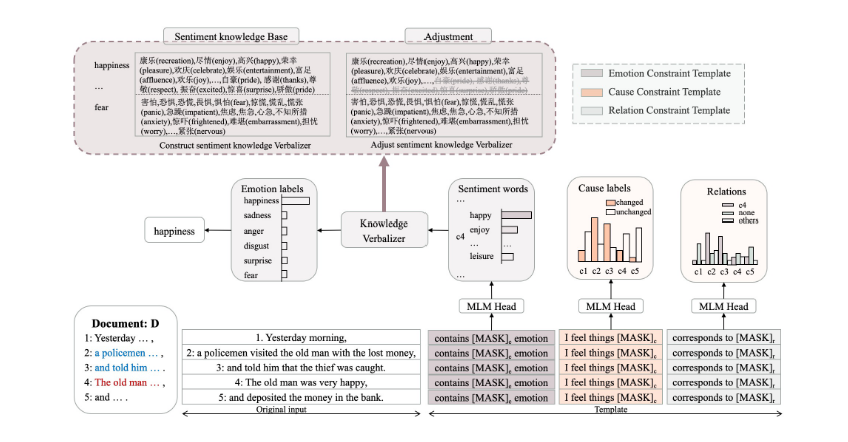
微调方法的不足：

之前的方法多为微调范例，也就是对每一个句子来说先从预训练模型或是词嵌入中获得每个句子的特征表示，然后上下文模块获得上下文表示，如果还需要，还有添加位置嵌入，或是和上文一样，通过分析数据集的设置特定的融合范围。最后得到表示进行全连接层的网络输出预测结果。但是不同的数据集就会有不同结果，通用性不够。

为了提出的第一个解决方法是，引入社会常识知识（好多论文都用这个方法），通过分析数据集发现，对每个句子进行xReact（同说话人的反应情绪）后得到的信息，如果一个真实的情绪句子的周围有不同的情绪产生，大概率就是原因句子所在位置。所以我们将原因句子识别任务转换为子提示：“I feel things changed / unchanged”以利用文本的潜在知识。而对于情绪句子识别任务，则是通过外部知识库的方式扩展标签词，将更多种词语归类为特定情绪，具体实现就是类似情感词典的方法，为六大情绪添加更多情绪词。同时为了解决扩展标签词带来的噪声，设计了一种基于相关性的选择方法。

模型组成：  
 本模型由四个部分组成，任务分解的子提示，情感语言器的构建，语言器调整和任务实现。整体流程是，先借助情感增强的提示调整块，将分类任务转换成掩蔽语言建模问题，然后根据提示模板将文本的输入转换成特定的表示，之后PLM获取词汇表中的每个单词被填充的[MASK]的概率，继而得到CE，ECPE任务的标签映射，对于EE任务则是使用预先构建的情感标签词集执行映射。

整体架构图如下：左下角的第二个为输入模块，第一个模块是结果，右下角三个为模板信息，分别是情绪识别，原因识别和情绪-原因对的识别任务，将他们输入MLM中得到结果。最上边为中英文的标签词库，因为EE任务得到的提示词可能并不是完全与六大标签对应，而是六大标签的投影，所以需要合理的对应关系让MASK的结果映射到合理的情绪上。

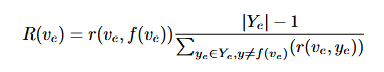


情绪对应词表的构造：

标签词的初始数据集会从外部词库中得到，但需要依据标签词与六大情感类别的相关性来筛选，为此需要计算标签词和情感类别之间的余弦相似度，为此需要得到他们的词向量表示。对于每个情感表示词的表示，流程如下：1、先将所有表示词都作为样本数据集中EE任务的提示MASK的label项，这样可以让输出的时候包含对于每个词表的向量。2、经过WoBERT-Plus的处理后，获取每一个情感句子的CLS标记的表示作为整个情感句子的表示，然后在词表中获取每一个情感标签词的token。3、遍历6种情绪，获取其所包含的情绪标签词的在所有情感句子中的概率，然后减去其平均值。4、然后将所有的情感标签词与每一个情绪类型的第一个标签词计算相似度，得到TF值。等价于每个情感标签词与六大情绪类型的相关度。

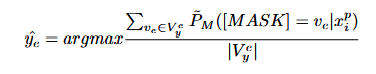


5.然后计算IDF（逆频率）值，即下面公式的右边部分，通过计算所有标签词与当前计算TF时的情绪不同的其他5种情绪的数值得到，具体的公式实现是[5（不同与当前TF计算使用的其他情绪类型个数）的1/(10/(6(总类型数) - 2 + 1e-2) + 1的次方)]/(当前情感标签词ve与其他5种情绪类型的相关度的1/(10/(6(总类型数) - 2 + 1e-2) + 1的次方)范数)得到，总之计算的结果还是它，就是实现方法有点怪怪的。最后将两者相乘得到TF-IDF分数。并且对于TF分数为负数的，就代表了完全不相关，直接记结果为-100。



情绪识别模板：

对每一个句子都进行判断是不是六大情绪之一，或者无明显情绪(中立)。模板是“contains [MASK] emotion”，输入一个子句后得到的MLM的输入就是“[CLS]4:文本.contains [MASK] emotion.”（如果是UniMEEC的话还会将输入文本信息进行词嵌入）然后得到MASK的结果会是一个由附加情感词组成的集合，然后可以将所有情绪类型对应的所有可选标签词，经过WoBERT模型后得到的概率进行求平均值，然后得到6种情绪类型，和无情绪的7种总的预测值，选择最大值，就是这个将句子预测的情绪类型了，具体的计算方式如下



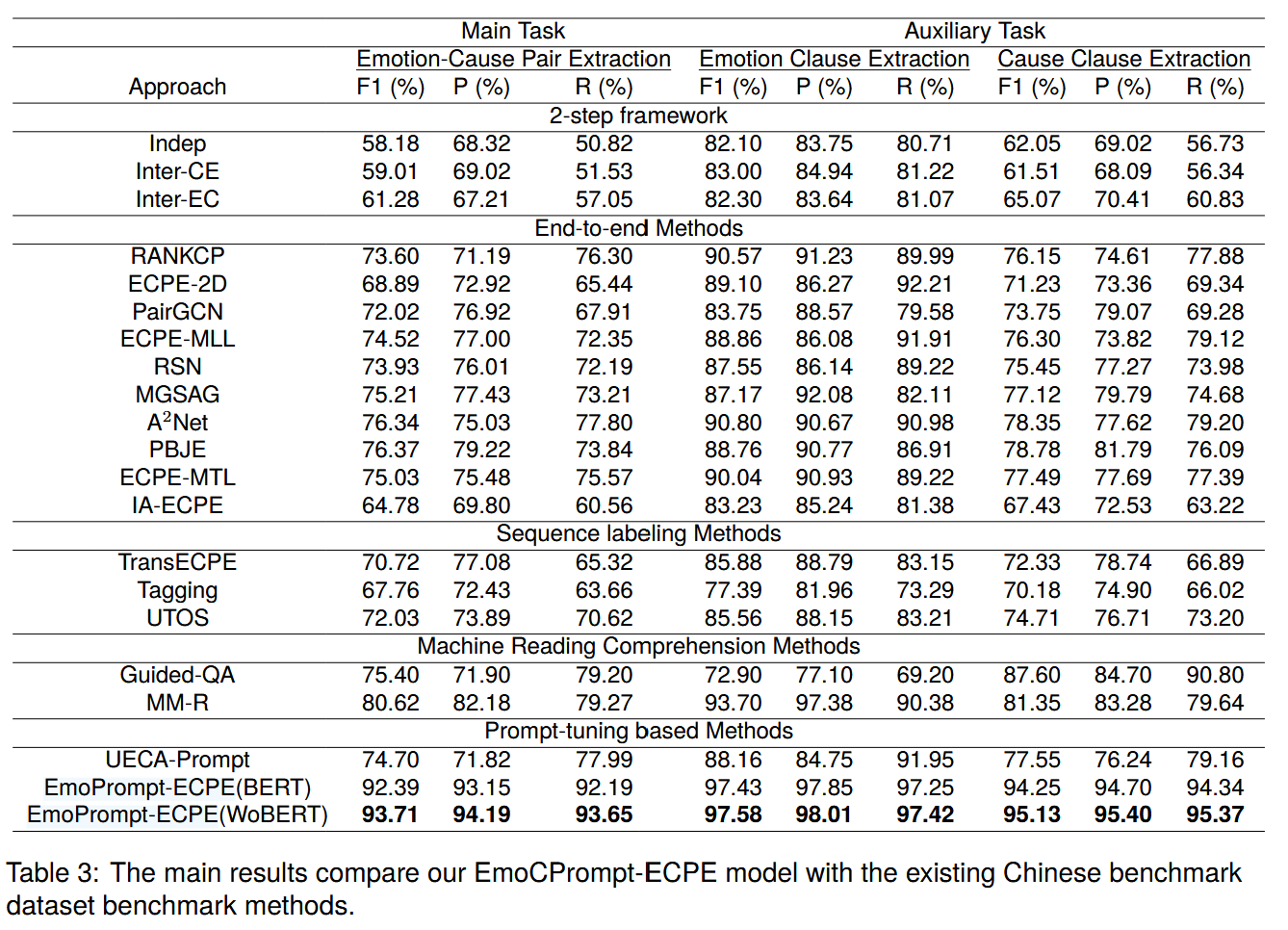
原因约束模板：

判断一个句子是不是原因句子，其模板是”“I feel things [MASK]”，预测答案的[MASK]的候选标签词包括 “changed” 和 “unchanged”。用于指示句子在读入前后文后，有没有出现情绪的转变。“changed” 表示当前子句是 cause 子句，而 “unchanged” 表示它不是。

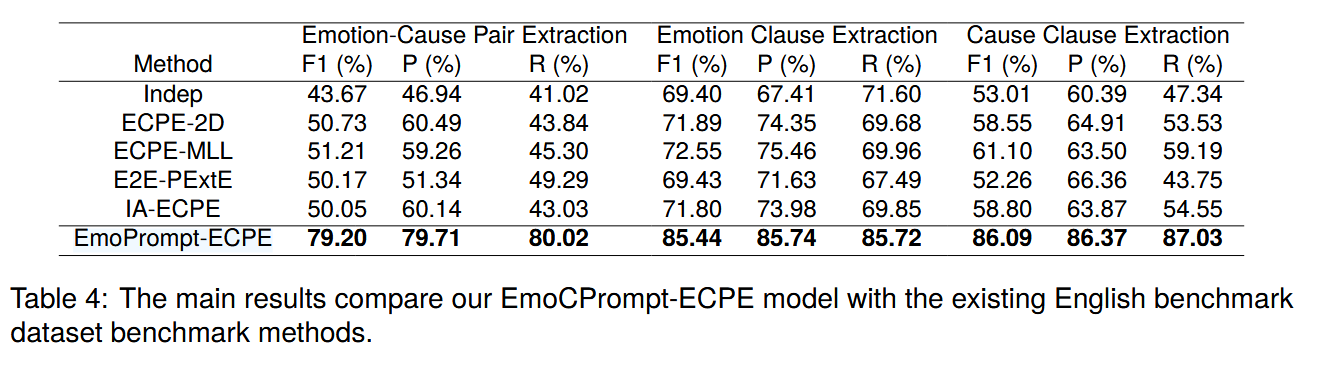
关系约束模块：  
 用于确定与当前子句关联的子句是什么，其模板为”correspond to [MASK]”，[MASK]的预选答案包括{1,2,...n,none}，其中n表示文档中的话语总数，none表示这个句子不与任何句子相关。此外为了让模型能够更好的了解文档中子句的顺序，需要文档中每个句子是第几个句子的数字标识也作为输入。（这个到是区别与UniMEEC了）。

整体的提示模板就是将上述三种模板直接合在一起，C[i]Mask[ee]Mask[ce]Mask[ECPE]。通过Mask[ee]计算得到句子的情绪类型，Mask[ce]得到句子是不是相较于前文发生了情绪转换，和Mask[ECPE]的结果得到的文档中的当前句子的那一句话可能是导致它的原因。计算每一个Mask的结果是根据每个提示模板句子经过WoBERT+模型后，取出其中所有可能选择的labels的概率最大值。然后对于每一个情绪句子i，如果Mask[ECPE]指示的句子的Mask[ce]是changed，就可以认为得到了一个情绪-原因对 （i，j），而其他的情况都认为不是一个情绪-原因对，不过实际操作就是依据每一句话的Mask结果直接与实际结果进行比对。其中Mask[EE]任务使用了FocalLoss函数另外两个任务使用交叉熵函数计算损失。

实验结果：  
 batch\_size = 2 学习率 1e-5，WoBERT+的大模型。在中文数据集上的结果如下，

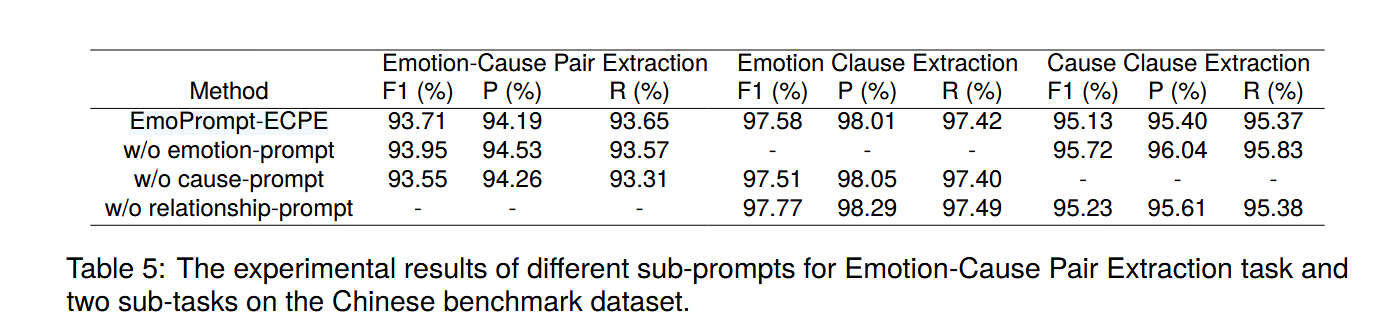


在情绪和原因从句的判断上得到了更好的预测值，并且WoBERT相较于BERT，词表从分字变成了分词，准确率也到了些微的提升。此外作为融合了整体的任务，相较于Inter-CE,EC任务这些多步模型来说，性能提升最显著。而作为端到端联合多任务学习方法的RANKCP这组方法来说，性能相较于多步模型也有提升，但是他们使用的仍是对三个子任务之间关系进行显式建模，比如就KEC模型来说，就是手动根据情绪-原因对，构建图网络并融合社会常识知识进行学习，与本模型提出使用构建子提示来对关系进行建模相比稍显逊色。与将 ECPE 任务当作序列标注问题的 TransECPE、Tagging 和 UTOS 方法，或将 ECPE 任务转换为机器阅读理解问题的 Guided - QA 和 MM - R 方法相比，我们的方法仍然具有竞争力。说明将ECPE任务作为掩码预测问题能更好的利用文本上下文信息，而此处再次提升的关键就是提示词的构建了，现有方法都是手动设计提示词，以针对不同数据集的情况，鲁棒性还稍逊。



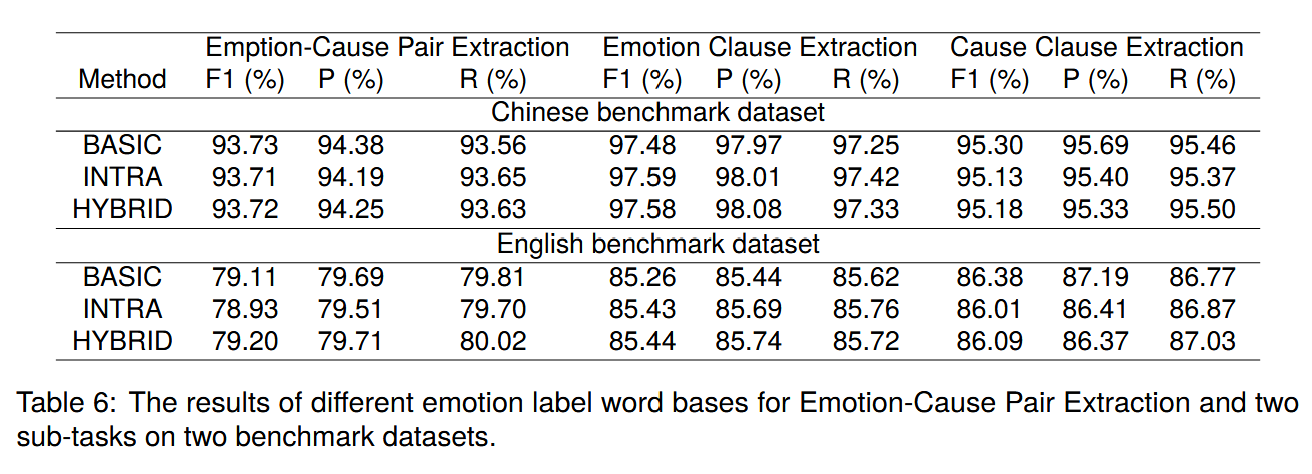
这一点在英文数据集的使用中就体现出来，随着数据集分布，或是中文提示词不能简单的翻译为英文进行提示的原因，模型精度虽然仍高于其他模型，但不如中文数据集上的良好表现（吐槽一下，它的源码好像删去了对英文的处理，可能需要重新分析标签词，句子id也可能需要换成英文的）。

此外，文章还分析了整合模型的重要性



如果将三个任务拆去一个子任务，模型精度也会有一些下降。最为特殊的是原因提示的消去对ECPE任务的结果影响最大，也间接证明了识别好原因句子才是ECPE任务的关键。

之后分析了不同情感词库，以获取不同的情感标签词



分析发现HYBRID的情感词库较优。