之前的讨论

本来是在整理一个比较长的笔记，这个只是其中一部分内容。其他的还没整理完，先把这个备忘发给老师看一下，后面的我稍晚一些会发过来。

第一次讨论：2018-07-23

这次是留完论文任务之后第一次打电话，忘记录音了。印象里这次讨论主要是大致汇报了一下看三篇论文的情况，将三篇论文简单整理成了笔记，捋顺了一下思路，确定了一个初步的idea，以及制定了后续讨论的计划（每周二下午三点）。

第二次讨论：2018-08-06

这次微信电话我进行了录音，方便后面回过头来听，回顾一下当时讨论了些啥。这次主要是复述了一下对问题的理解。对比了snowball工作与COMET工作的可比性、异同。老师的答复是：多走了一步

COMET：利用人工标注的Transformer模型进行迁移学习，使其迁移到生成常识上来

snowball：初始有一些种子，在语料库里不断迭代，也能发现许多不同的关系

需要思考的是：snowball和COMET这两个模型本质的异同。

snowball里面隐含的是：self-supervised的思想。不依靠人工标注。用余弦相似度，设定某一阈值，高于某一阈值的补充进来，不断地滚。目前的研究套路大多也是这样，利用embedding空间里的夹角余弦值，补充进来，relation也可以做embedding。通过某种方式发现新的关系，然后在关系的空间里分分类，设定一些评估，然后不断的滚，也可以实现目的。

Transformer线的思想：为什么用这样的预训练模型也可以发现关系，而且结果是靠谱的，他背后的本质是什么？

snowball本质：有一个自监督机制，里面有一个对抗生成的思路，通过监督，使得更好的样本能够进来，不断地自增长。

用“两条腿”来理解——从自监督的角度来理解Transformer，从Transformer的角度来理解自监督学习，最后看看能不能把它们融合在一起，然后可以做的比原来更好一点，比如可以引入人工？因为用Transformer就不能引用人工了，不能引入专家知识。在这个过程中假如可以引入专家知识，达到更大的效果，也是很好的。用到的知识面很宽广，可以用到生成对抗（GAN），也有自监督的思路，跟Transformer也有关系，把这些线都搞清楚。先以知识为主，把这两条线都走通，先学的很清晰，然后再讨论研究。

后面又讨论了一些关于长线研究和短线、快速占坑的话题。提到了夏令营座谈会那次那个发ACL的同学混淆“数据增强”和“对抗样本”的问题。突然有一点感受就是，做科研还是应该先沉淀许多知识，然后在idea的层面上多思考思考问题的本质，后面再关注具体实现的问题。而且由于老师这边压力不是很大，不急着出成果，也就有了更多时间用来学习、思考，我个人认为这一点我以后会很受益。谈到想idea，感觉自己数学基础知识差的确实有点大，就上次讨论时曾经提到过“EM算法”、“核函数”、“随机过程”等等概念，我回去查了一下，发现需要学习《随机过程》、《统计学习方法》等等。本科阶段也只是学了高等数学、线性代数、概率统计、数值分析这四门课，会的也都是一些基础的东西，感觉我这些数学基本功不深的话，对问题很难有一个全面的理解。不知道研究生会不会有这些课程，或者需要自学一下某些数学课。感觉自己需要补一下数学知识。（老师能不能给列几门课程，我回去学一下？）

后面说到，目前生命科学那边的一个分支也是自监督的线，学出一个知识图谱，在其上做推理等等。目前还是先学习，把这些基本概念都搞清，等到差不多年末的时候也可以有准备的进来。

COMET是利用Transformer这个**语言模型**，用task训练使其带着语义信息迁移过来，相当于snowball里面的哪一步？——snowball通过现有的tuple去语料库里"学得"上下文vector，再通过这些vector去从语料库里生成新的tuple。这里的vector其实就相当于在这个问题下抽象出来的，能够生成tuple的"**语言模型**"。他的测试相当于人工监督，去筛掉一些不好的样本，而机器学习里面相当于是利用最小化loss，优化模型参数。一个是人工监督，一个是非人工干预。snowball相当于是一轮一轮的在训练，而数据是增量的。

这两个工作各有其长短的地方。Transformer这个语言模型本身非常复杂，但是机制比snowball简单；snowball的过程有它领先的意义，是一种自监督的、增量的训练。我们的目的就是在这两个点中间夹出一个新的点。

第三次讨论：2018-08-13

这周回去主要是沿着Transformer和Pre-trained模型这条线捋的。首先解释了一下Transformer的原理。

第一个问题：什么叫Attention？详细讨论了一下Attention机制。相当于是一个衡量词之间重要程度的关系矩阵，然后这个矩阵是可以被学出来的。这里提到，了解一个机制，一是了解其原理本身，二是其怎么实现的。

第二个问题：什么叫Multi-Headed Attention。为什么要Multi-Headed。为什么8个Head随机初始化之后可以分别关注到不同方面的特征？这里提到了EM算法。（我回去查了查，还没弄懂。。）分成Multi-Head以后，维度变小了，而且还能表达不同的方面。一是为什么分成了8个Head之后维度会小？二是它是怎么样被建模的？为什么每一个子空间是独立的，都能学出来？而且学出来的东西都是不一样的？直觉是，其借用了EM算法大体的一个思路。把每一个相互独立的特征空间，特征跟特征之间是有比较强的耦合的，是一个联合概率分布，一个联合概率分布就等价于一个子空间。一个联合概率分布是一个特征集。"高斯混合模型"。每一个特征集都是一个耦合比较紧密的子空间。在学习的过程中多个Head综合在一起会有一个互斥的作用。在一个Head编码了这块信息之后，其他Head就会被挤掉，就不用编码这一块信息。"狄利克雷过程"。

后面是说具体实现。首先说了一下Query、Key和Value三个矩阵是怎么来的。上次由于没提前准备文字材料，直接口头描述比较抽象。我接下来会整理一份Multi-Head Attention和Transformer的材料出来，提前发给老师。

提到概念：张量分解。

后面又讨论Transformer的encode、Decode的问题。我自己Transformer解码的地方还没太搞清楚。

后面又提了我顺着pre-trained语言模型这一条线看的一些内容。从Word2Vec——ELMo——GPT——再到BERT。

最后我提了关于COMET里面的一个疑问，到底是输入s和r的拼接，o作为标签，还是sro都拼接到一起作为输入，这样的话标签是什么？后来讨论之后发现是理解插图理解的有偏差，图示的意思是数据格式的编码不同，和他模型的输入没关系。

提到：自编码器——Auto Encoder（回去补）

这周主要是学Transformer的细节以及了解Pre-trained模型的思想。老师希望先把目光放在思路上，对问题有一个整体的把握，之后再下去关注这些实现的细节。先关注这个思路有没有人做过研究。因此这周回去是理解自监督、看看自监督在NLP里面有什么应用，以及和我们的问题（snowball算法，关系抽取）如何结合到一起。用语言模型生成出一些新的relation，再把这些新生成的relation放到语言模型里tune一tune，然后再生成新的relation。