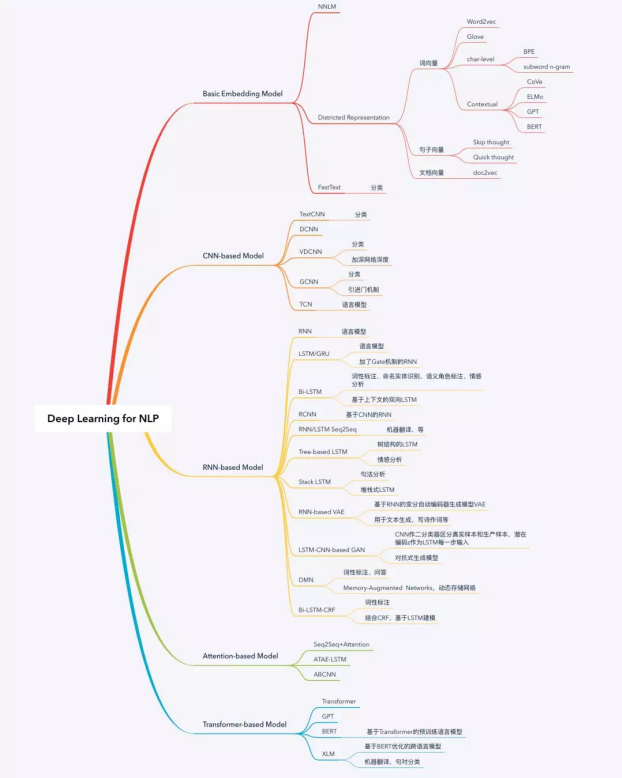
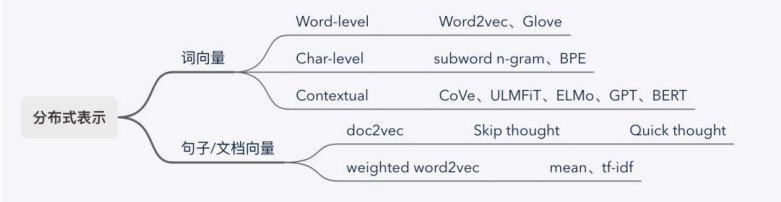
nlp面试准备

- 三件事：知识图谱、nlp模型总结、linux的基本操作



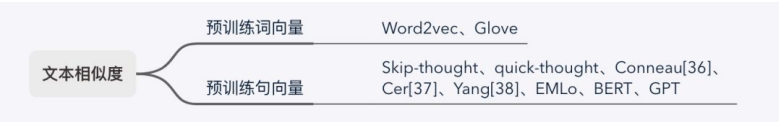
- **向量表示**
  - 词、句子、文章的向量表示，包括基于上下文的方法。词的向量表示，看任务对词粒度的要求，一般场景用预训练好的词向量Word2vec和Glove，或者通过one-shot编码再加一个Embedding层的方式。在序列标注、机器翻译等场景，对词内特征要求比较高，可以采用字级别的，常用的有subword n-gram和BPE，也有结合词级别向量的做法。迁移学习带来的模型效果提升有目共睹，EMLo、GPT、BERT等预训练模型，将上下文考虑进来，在具体下游任务微调适配，更恰当的表示词的语义信息，特别是一词多义场景。句子和文档等序列的向量表示，大概分两类：一种是用TextRank的方式提取序列的主要关键词，基于关键词的词向量并赋予权重的方式表示序列，权重分配可以是平均也可以用tf-idf之类的算法；一种是将序列进行embedding，方法有很多种，譬如doc2vec、skip-thought、quick-thought、BERT等。



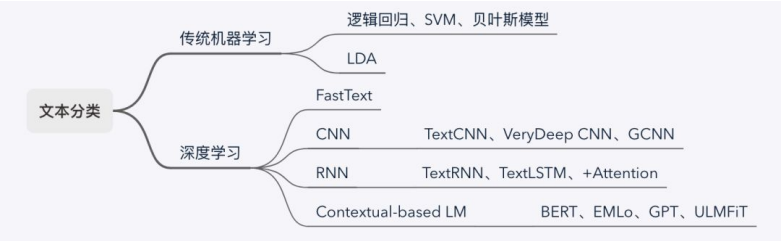
  - TF-IDF可以没有词向量转化的概念
    - 实际上就是tf*idf的值，来表示该词对于这个文本的重要性。
  - word2vec
    - 模型框架根据输入输出不同，主要包括CBOW和skip-gram，也就是说这是两种模型

    - 实际上，word2vec是一个简单的三层线性神经网络：输入、隐藏、输出；实现就是训练两个共享参数矩阵，也称为“fake task”，因为我们不需要用到这个模型的输出层，而需要该模型的隐藏层，对文本进行编码。
    - word2vec有三种优化方式：
        - hierarchical softmax（层次softmax）的输出为huffman Tree
            - 因为要对输出层做softmax，对于词空间数十万的数量，所需计算是在太大了，所以用哈夫曼树作为输出层，将词频高的路径缩短，在计算softmax上节约了大量的算力。实际上将V分类的问题，转化成log（ｖ）次的二分类。
            - huffman Tree（哈夫曼树）
        - negative sample（负采样）负采样则不使用哈夫曼树，对于随机选取指定数量的负样本和正样本进行参数更新，例如输出为one-hot，则期盼输出为0的称为负样本，期盼输出为1的称为正样本。
- fasttext
    - fasttext可以说是word2vec的衍生，不同的是在word embedding方面，fasttext采用字符级别的n-gram概率，例如apple，会加入起始符号和结束符号，然后根据n将apple分为<ap, app, ppl, ple, le>然后得到n-gram的向量作为模型输入。
    - 该模型输入到隐藏层未使用激活函数，但是隐藏到输出使用了激活函数，输出的时候采用了分层softmax
- Elmo
    - 模型由两层Bi-LSTM构成的，该双向LSTM是由大量的语料训练得来的，LSTM之间有残差链接，作者认为低层LSTM能够提取句法信息，高层LSTM能够提取语义信息。
    - 实际上训练Bi-LSTM才是该模型的主要任务，然后该任务输出可以作为其他模型word embedding的输入。
    - 他解决word2vec和fasttext的静态Embedding的问题，对于特定语境能够识别出一词多义。
- BERT
    - word Embeding：将每一个词通过字向量查询表转化为字向量，而这个字向量查询表是通过word pieces的算法构成的。
    - segmen Embedding：将不同句子中间方法一个【sep】的标识符，
    - position embedding：对位置进行编码，因为不同位置可能所代表的意思就不相同。
- RoBerta
    - 和bert不同点：
        - 更大的数据量，原本bert只有16G的样本数据，robert的数据量接近150G，近十倍的数据量
        - 更大的batch_size，原本bert的batch_size只有256，robert的batch_size增加到了2k，8k

        - 更长的训练时间
        - 动态mask，bert是数据输入模型时候就将数据mask好，为了充分利用数据，那么bert对这些数据进行复制，那么这些复制的数据并不是只在一个epoch输入，而不是不同的epoch，那么就会造成epoch训练相同的mask的数据。roberta是将数据输入模型后再进行随机mask。中文还没实现动态mask。
        - no NSP and input format：实际上没说为啥，实验对比的跑分，no NSP稍微高点
    - roberta和bert 的text embedding是一样的，不一样的是wordpiece的方式不同：
        - BERT原型使用的是BPE , RoBERTa使用了GPT2的 byte BPE 实现，使用的是byte而不是unicode characters作为subword的单位。
- GPT2
    - GPT-2也是从嵌入矩阵中查找到对应的词向量 + 位置编码
    - 类似于RNN输出是一个一个输出的，也就是AR模型
- XLNet
    - 尝试解决AR模型和AE模型所存在的问题，将语言排列组合，算概率
    - AR的问题是单向性
    - AE的问题是预训练是使用了【mask】和微调是没有，这会导致数据不一致。
- **文本相似性**
- 大家应该很熟悉谷歌提出的simhash算法，用来文本去重。深度学习用来处理文本语义相似性任务，简单可以分成两类：一种是用预训练好的词向量Word2vec或Glove来表示句子，计算向量之间的距离来区分相似性；一种是用预训练模型来表示句子，譬如2018年谷歌[37]提出通用句子编码器来获取句子向量，然后用arccos来计算相似距离，作者在论文中介绍了两种编码器：DAN和Tansformer。
    - **文本分类**
-



- 文本分类，NLP工业化应用最广泛的任务之一，譬如辨别垃圾信息或恶意评论、对文章进行政治倾向分类、对商品积极和消极的评论进行分类，等等。文本分类的方法有很多种，传统机器学习的逻辑回归、SVM、贝叶斯分类模型、主题模型，深度学习的FastText、基于CNN/RNN的分类模型，以及最近很火的基于预训练模型的BERT、ELMo、GPT、ULMFiT。如果是语料语义简单的分类任务，用传统机器学习方法即可，抑或FastText也是个不错的选择，成本低，如果是像社交数据之类语义丰富的场景，可以考虑深度学习模型，CNN擅长捕获局部特征，RNN擅长处理时序信息，预训练模型的优点就不用多说，注意力机制在模型效果不满足时可以考虑一试。
-



- **情感分析**
- 情感分析，又叫观点挖掘，该任务目的是从文本中研究人们对实体以及其属性所表达的观点、情绪、情感、评价和态度。这些实体可以是各种产品、机构、服务、个人、事件、问题或主题等。这一领域涉及的问题十分多样，包括很多研究任务，譬如情感分析、观点挖掘、观点信息提取、情感挖掘、主观性分析、倾向性分析、情绪分析以及评论挖掘等。基于所处理文本的颗粒度，情感分析研究可以分成三个级别：篇章级、句子级和属性级。研究情感分析的方法有很多种，情感词典匹配规则、传统机器学习、深度学习，等。
-

- **机器翻译**
- 机器翻译，非常具有挑战性的NLP任务之一。从基于短语匹配概率的SMT框架，到基于CNN/RNN/Transformer的NMT框架，注意力机制在提升模型效果发挥重要的作用。机器翻译任务里，低资源语言的翻译问题是个大难点，跨语言模型XLM在这块进行了探索，利用富资源语言来学习低资源语言。
-

- **命名实体识别**
- 命名实体识别（英语：Named Entity Recognition，简称NER）是NLP序列标注任务的一种，指从输入文本中识别出有特定意义或指代性强的实体，是机器翻译、知识图谱、关系抽取、问答系统等的基础。学术上NER的命名实体分3大类和7小类，3大类指实体类、时间类、数字类，7小类指人名、地名、组织机构名、时间、日期、货币、百分比。语言具有语法，语料遵循一定的语法结构，所以CRF、HMM和MEMM等概率图模型被用来分析标签转移概率，包括深度学习模型一般会加上CRF层来负责句子级别的标签预测。深度学习模型一般用ID-CNN和Bi-LSTM再加一个CRF层，迁移学习火起来后，CVT、ELMo和BERT在NER任务上的表现也是非常不错。
-

以上内容整理于 幕布文档