【关于 NLP】 那些你不知道的事

作者：杨夕

项目地址：https://github.com/km1994/nlp\_paper\_study

个人介绍：大佬们好，我叫杨夕，该项目主要是本人在研读顶会论文和复现经典论文过程中，所见、所思、所想、所闻，可能存在一些理解错误，希望大佬们多多指正。

目录

* [【关于 NLP】 那些你不知道的事](#%E5%85%B3%E4%BA%8E-nlp-%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
  + [目录](#%E7%9B%AE%E5%BD%95)
  + [介绍](#%E4%BB%8B%E7%BB%8D)
    - [【关于 论文工具】那些你不知道的事](#%E5%85%B3%E4%BA%8E-%E8%AE%BA%E6%96%87%E5%B7%A5%E5%85%B7%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
    - [会议收集篇](#%E4%BC%9A%E8%AE%AE%E6%94%B6%E9%9B%86%E7%AF%87)
    - [NLP 学习篇](#nlp-%E5%AD%A6%E4%B9%A0%E7%AF%87)
      * [经典会议论文研读篇](#%E7%BB%8F%E5%85%B8%E4%BC%9A%E8%AE%AE%E8%AE%BA%E6%96%87%E7%A0%94%E8%AF%BB%E7%AF%87)
        + [理论学习篇](#%E7%90%86%E8%AE%BA%E5%AD%A6%E4%B9%A0%E7%AF%87)
      * [经典论文研读篇](#%E7%BB%8F%E5%85%B8%E8%AE%BA%E6%96%87%E7%A0%94%E8%AF%BB%E7%AF%87)
      * [【关于 transformer 】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-transformer--%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 预训练模型】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E9%A2%84%E8%AE%AD%E7%BB%83%E6%A8%A1%E5%9E%8B-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 信息抽取】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E4%BF%A1%E6%81%AF%E6%8A%BD%E5%8F%96-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【关于 实体关系联合抽取】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E8%81%94%E5%90%88%E6%8A%BD%E5%8F%96-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【关于 命名实体识别】那些你不知道的事：](#%E5%85%B3%E4%BA%8E-%E5%91%BD%E5%90%8D%E5%AE%9E%E4%BD%93%E8%AF%86%E5%88%AB%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【关于 关系抽取】那些你不知道的事：](#%E5%85%B3%E4%BA%8E-%E5%85%B3%E7%B3%BB%E6%8A%BD%E5%8F%96%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【关于 文档级别关系抽取】那些你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%96%87%E6%A1%A3%E7%BA%A7%E5%88%AB%E5%85%B3%E7%B3%BB%E6%8A%BD%E5%8F%96%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 知识图谱 】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1--%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【实体链指篇】](#%E5%AE%9E%E4%BD%93%E9%93%BE%E6%8C%87%E7%AF%87)
        + [【【关于 实体消歧 】 那些的你不知道的事】](#%E5%85%B3%E4%BA%8E-%E5%AE%9E%E4%BD%93%E6%B6%88%E6%AD%A7--%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【【关于KGQA 】 那些的你不知道的事】](#%E5%85%B3%E4%BA%8Ekgqa--%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [【【关于Neo4j 】 那些的你不知道的事】](#%E5%85%B3%E4%BA%8Eneo4j---%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 细粒度情感分析】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E7%BB%86%E7%B2%92%E5%BA%A6%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 主动学习】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E4%B8%BB%E5%8A%A8%E5%AD%A6%E4%B9%A0-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 对抗训练】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E5%AF%B9%E6%8A%97%E8%AE%AD%E7%BB%83-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 GCN in NLP 】那些你不知道的事：](#%E5%85%B3%E4%BA%8E-gcn-in-nlp-%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 文本预处理】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%96%87%E6%9C%AC%E9%A2%84%E5%A4%84%E7%90%86-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于问答系统】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E%E9%97%AE%E7%AD%94%E7%B3%BB%E7%BB%9F-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 文本摘要】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%96%87%E6%9C%AC%E6%91%98%E8%A6%81-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 文本匹配】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%96%87%E6%9C%AC%E5%8C%B9%E9%85%8D-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 机器翻译】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%9C%BA%E5%99%A8%E7%BF%BB%E8%AF%91-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 文本生成】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E6%96%87%E6%9C%AC%E7%94%9F%E6%88%90-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 实体消歧】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E5%AE%9E%E4%BD%93%E6%B6%88%E6%AD%A7-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 对话系统】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E5%AF%B9%E8%AF%9D%E7%B3%BB%E7%BB%9F-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 Rasa 】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-rasa--%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 半监督学习】 那些的你不知道的事](#%E5%85%B3%E4%BA%8E-%E5%8D%8A%E7%9B%91%E7%9D%A3%E5%AD%A6%E4%B9%A0-%E9%82%A3%E4%BA%9B%E7%9A%84%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
      * [【关于 NLP分类任务】那些你不知道的事](#%E5%85%B3%E4%BA%8E-nlp%E5%88%86%E7%B1%BB%E4%BB%BB%E5%8A%A1%E9%82%A3%E4%BA%9B%E4%BD%A0%E4%B8%8D%E7%9F%A5%E9%81%93%E7%9A%84%E4%BA%8B)
        + [实战篇](#%E5%AE%9E%E6%88%98%E7%AF%87)
      * [重点推荐篇](#%E9%87%8D%E7%82%B9%E6%8E%A8%E8%8D%90%E7%AF%87)
      * [推荐篇](#%E6%8E%A8%E8%8D%90%E7%AF%87)
        + [视频学习篇](#%E8%A7%86%E9%A2%91%E5%AD%A6%E4%B9%A0%E7%AF%87)
    - [Elastrsearch 学习篇](#elastrsearch-%E5%AD%A6%E4%B9%A0%E7%AF%87)
    - [推荐系统 学习篇](#%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F-%E5%AD%A6%E4%B9%A0%E7%AF%87)
    - [竞赛篇](#%E7%AB%9E%E8%B5%9B%E7%AF%87)
    - [GCN\_study学习篇](#gcn_study%E5%AD%A6%E4%B9%A0%E7%AF%87)
    - [ML 小白入门篇](#ml-%E5%B0%8F%E7%99%BD%E5%85%A5%E9%97%A8%E7%AF%87)
    - [资源篇](#%E8%B5%84%E6%BA%90%E7%AF%87)
    - [CV 入门 实战篇](#cv-%E5%85%A5%E9%97%A8-%E5%AE%9E%E6%88%98%E7%AF%87)

介绍

[【关于 论文工具】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/other_study/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea/)

* 问题
  + 作为一名 scholar，你是否和我一样，在刚入门 NLP 时，对于陌生领域有种无从下手，心存畏惧？
  + 作为一名 scholar，你是否还在发愁如何找好的论文？
  + 作为一名 scholar，你是否还在为 自己 的 英文阅读 能力跟不上 很烦恼？
  + 作为一名 scholar，你是否还在为 看到 一篇好paper，但是复现不出 code 而心累？
  + 作为一名 scholar，你是否还在为 有Good idea，Outstanding Experimental results，Beautiful Chinese manuscript，结果 Bad English manuscript, Poor Journal 而奔溃？
  + 作为一名 scholar，你是否在为搞科研没人交流而自闭？
* 当你看到这一篇文档，你将不在为这些问题而烦恼，因为我们为你准备了一整套免费的从 论文查找->论文翻译->论文理解->相关代码搜索->写英文稿->科研学术交流 的路径。
  + [论文不会找怎么办？](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E8%AE%BA%E6%96%87%E4%B8%8D%E4%BC%9A%E6%89%BE%E6%80%8E%E4%B9%88%E5%8A%9E)
    - [顶会资讯](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E9%A1%B6%E4%BC%9A%E8%B5%84%E8%AE%AF)
    - [论文搜索和分析工具](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E8%AE%BA%E6%96%87%E6%90%9C%E7%B4%A2%E5%92%8C%E5%88%86%E6%9E%90%E5%B7%A5%E5%85%B7)
  + [外文读不懂怎么办？](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E5%A4%96%E6%96%87%E8%AF%BB%E4%B8%8D%E6%87%82%E6%80%8E%E4%B9%88%E5%8A%9E)
    - [论文翻译神器 ———— 通天塔](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E8%AE%BA%E6%96%87%E7%BF%BB%E8%AF%91%E7%A5%9E%E5%99%A8--%E9%80%9A%E5%A4%A9%E5%A1%94)
    - [论文翻译小助手 ———— 彩云小译](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E8%AE%BA%E6%96%87%E7%BF%BB%E8%AF%91%E5%B0%8F%E5%8A%A9%E6%89%8B--%E5%BD%A9%E4%BA%91%E5%B0%8F%E8%AF%91)
  + [外文没 code 怎么办？](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E5%A4%96%E6%96%87%E6%B2%A1-code-%E6%80%8E%E4%B9%88%E5%8A%9E)
    - [papers with code](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#papers-with-code)
    - [OpenGitHub 新项目快报](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#opengithub-%E6%96%B0%E9%A1%B9%E7%9B%AE%E5%BF%AB%E6%8A%A5)
  + [外文写起来麻烦怎么办](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E5%A4%96%E6%96%87%E5%86%99%E8%B5%B7%E6%9D%A5%E9%BA%BB%E7%83%A6%E6%80%8E%E4%B9%88%E5%8A%9E)
    - [Overleaf](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#overleaf)
    - [Authorea](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#authorea)
    - [Code ocean](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#code-ocean)
  + [搞科研没人交流怎么办？](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#%E6%90%9E%E7%A7%91%E7%A0%94%E6%B2%A1%E4%BA%BA%E4%BA%A4%E6%B5%81%E6%80%8E%E4%B9%88%E5%8A%9E)
    - [Shortscience](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#shortscience)
    - [OpenReview](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#openreview)
    - [Scirate](https://github.com/km1994/nlp_paper_study/tree/master/%E8%AE%BA%E6%96%87%E5%AD%A6%E4%B9%A0idea#scirate)

会议收集篇

* [ACL2020](ACL/ACL2020.md)
* [SIGIR2020](SIGIR_stduy/readme.md/#sigir-2020)

NLP 学习篇

经典会议论文研读篇

* [ACL2020](ACL/ACL2020.md)
  + [【关于 CHECKLIST】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/ACL2020_bertpaper_CHECKLIST/)
    - 阅读理由：ACL2020 best paper ，利用 软件工程 的 思想 思考 深度学习
    - 动机：针对 train-val-test 分割方法 评估 模型性能容易出现 不全面、偏向性、可解性差问题；
    - 方法：提出了一种模型无关和任务无关的测试方法checklist，它使用三种不同的测试类型来测试模型的独立性。
    - 效果：checklist揭示了大型软件公司开发的商业系统中的关键缺陷，表明它是对当前实践的补充好吧。测试使用 checklist 创建的模型可以应用于任何模型，这样就可以很容易地将其纳入当前的基准测试或评估中管道。

理论学习篇

经典论文研读篇

* 那些你所不知道的事
  + [【关于Transformer】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/transformer_study/Transformer/)
  + [【关于Bert】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/T1_bert/)

【关于 transformer 】 那些的你不知道的事

* [【关于Transformer】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/) transformer 论文学习
  + [【关于Transformer】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/Transformer/)
    1. 为什么要有 Transformer?
    2. Transformer 作用是什么？
    3. Transformer 整体结构怎么样？
    4. Transformer-encoder 结构怎么样？
    5. Transformer-decoder 结构怎么样?
    6. 传统 attention 是什么?
    7. self-attention 长怎么样?
    8. self-attention 如何解决长距离依赖问题？
    9. self-attention 如何并行化？
    10. multi-head attention 怎么解?
    11. 为什么要 加入 position embedding ？
    12. 为什么要 加入 残差模块？
    13. Layer normalization。Normalization 是什么?
    14. 什么是 Mask？
    15. Transformer 存在问题？
    16. Transformer 怎么 Coding?
  + [【关于 Transformer-XL】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/T3_Transformer_XL/)
    - 动机
      * RNN：主要面临梯度消失或爆炸（gradient vanishing and explosion），解决方法集中在优化方法、初始化策略、辅助记忆单元的研究上。
      * vanilla Transformer：最长建模长度是固定的，无法捕捉更长依赖关系；等长输入序列的获取通常没有遵循句子或语义边界（出于高效考虑，往往就是将文本按长度一段段截取，而没有采用padding机制），可能造成上下文碎片化（context fragmentation）。
    - 方法
      * 引入循环机制（Reccurrence，让上一segment的隐含状态可以传递到下一个segment）：将循环（recurrence）概念引入了深度自注意力网络。不再从头计算每个新segment的隐藏状态，而是复用从之前segments中获得的隐藏状态。被复用的隐藏状态视为当前segment的memory，而当前的segment为segments之间建立了循环连接（recurrent connection）。因此，超长依赖性建模成为了可能，因为信息可以通过循环连接来传播。
      * 提出一种新的相对位置编码方法，避免绝对位置编码在循环机制下的时序错乱：从之前的segment传递信息也可以解决上下文碎片化的问题。更重要的是，本文展示了使用相对位置而不是用绝对位置进行编码的必要性，这样做可以在不造成时间混乱（temporal confusion）的情况下，实现状态的复用。因此，作为额外的技术贡献，文本引入了简单但有效的相对位置编码公式，它可以泛化至比在训练过程中观察到的长度更长的注意力长度。
  + [【关于 SHA\_RNN】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/SHA_RNN_study/)
    - 论文名称：Single Headed Attention RNN: Stop Thinking With Your Head 单头注意力 RNN: 停止用你的头脑思考
  + [【关于 Universal Transformers】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/T4_Universal_Transformers/)
  + [【关于Style\_Transformer】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/Style_Transformer/LCNQA/)
  + [【关于 Linformer 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/ACL2020_Linformer)
    - 论文标题：《Linformer: Self-Attention with Linear Complexity》
    - 来源：ACL 2020
    - 链接：https://arxiv.org/abs/2006.04768
    - 参考：https://zhuanlan.zhihu.com/p/149890569
  + [【关于 Performer 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/transformer_study/Performer) **【推荐阅读】**
    - 阅读理由：Transformer 作者 Krzysztof Choromanski 针对 Transformer 问题的重新思考与改进
    - 动机：Transformer 有着巨大的内存和算力需求，因为它构造了一个注意力矩阵，需求与输入呈平方关系;
    - 思路：使用一个高效的（线性）广义注意力框架（generalized attention framework），允许基于不同相似性度量（核）的一类广泛的注意力机制。
    - 优点：该方法在保持线性空间和时间复杂度的同时准确率也很有保证，也可以应用到独立的 softmax 运算。此外，该方法还可以和可逆层等其他技术进行互操作。

【关于 预训练模型】 那些的你不知道的事

* [【关于Bert】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/)：Bert论文研读
  + [【关于Bert】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/T1_bert/)
    - 阅读理由：NLP 的 创世之作
    - 动机：word2vec 的多义词问题 && GPT 单向 Transformer && Elmo 双向LSTM
    - 介绍：Transformer的双向编码器
    - 思路：
      * 预训练：Task 1：Masked LM && Task 2：Next Sentence Prediction
      * 微调：直接利用 特定任务数据 微调
    - 优点：NLP 所有任务上都刷了一遍 SOTA
    - 缺点：
      * [MASK]预训练和微调之间的不匹配
      * Max Len 为 512
  + [【关于 XLNet 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/T2_XLNet/)
    - 阅读理由：Bert 问题上的改进
    - 动机：
      * Bert 预训练和微调之间的不匹配
      * Bert 的 Max Len 为 512
    - 介绍：广义自回归预训练方法
    - 思路：
      * 预训练：
        + Permutation Language Modeling【解决Bert 预训练和微调之间的不匹配】
        + Two-Stream Self-Attention for Target-Aware Representations【解决PLM出现的目标预测歧义】
        + XLNet将最先进的自回归模型Transformer-XL的思想整合到预训练中【解决 Bert 的 Max Len 为 512】
      * 微调：直接利用 特定任务数据 微调
    - 优点：
    - 缺点：
  + [【关于 RoBERTa】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/T4_RoBERTa/)
    - 阅读理由：Bert 问题上的改进
    - 动机：
      * 确定方法的哪些方面贡献最大可能是具有挑战性的
      * 训练在计算上是昂贵的的，限制了可能完成的调整量
    - 介绍：A Robustly Optimized BERT Pretraining Approach
    - 思路：
      * 预训练：
        + 去掉下一句预测(NSP)任务
        + 动态掩码
        + 文本编码
      * 微调：直接利用 特定任务数据 微调
    - 优点：
    - 缺点：
  + [【关于 ELECTRA 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/ELECTRA/)
    - 阅读理由：Bert 问题上的改进 【不推荐阅读，存在注水！】
    - 动机：
      * 只有15%的输入上是会有loss
    - 介绍：判别器 & 生成器 【但是最后发现非 判别器 & 生成器】
    - 思路：
      * 预训练：
        + 利用一个基于MLM的Generator来替换example中的某些个token，然后丢给Discriminator来判别
      * 微调：直接利用 特定任务数据 微调
    - 优点：
    - 缺点：
  + [【关于 Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/ACL2020_UnsupervisedBert/)
    - 论文链接：https://arxiv.org/pdf/2004.14786.pdf
    - 代码链接：https://github.com/bojone/perturbed\_masking
    - 动机
      * 通过引入少量的附加参数，probe learns 在监督方式中使用特征表示（例如，上下文嵌入）来 解决特定的语言任务（例如，依赖解析）。这样的probe tasks 的有效性被视为预训练模型编码语言知识的证据。但是，这种评估语言模型的方法会因 probe 本身所学知识量的不确定性而受到破坏
    - Perturbed Masking
      * 介绍：parameter-free probing technique
      * 目标：analyze and interpret pre-trained models，测量一个单词xj对预测另一个单词xi的影响，然后从该单词间信息中得出全局语言属性（例如，依赖树）。
    - 整体思想很直接，句法结构，其实本质上描述的是词和词之间的某种关系，如果我们能从BERT当中拿到词和词之间相互“作用”的信息，就能利用一些算法解析出句法结构。
  + [【关于 GRAPH-BERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/bert_study/T2020_GRAPH_BERT))
    - 论文名称：GRAPH-BERT: Only Attention is Needed for Learning Graph Representations
    - 论文地址：https://arxiv.org/abs/2001.05140
    - 论文代码：https://github.com/jwzhanggy/Graph-Bert
    - 动机
      * 传统的GNN技术问题：
        + 模型做深会存在suspended animation和over smoothing的问题。
        + 由于 graph 中每个结点相互连接的性质，一般都是丢进去一个完整的graph给他训练而很难用batch去并行化。
    - 方法：提出一种新的图神经网络模型GRAPH-BERT (Graph based BERT)，该模型只依赖于注意力机制，不涉及任何的图卷积和聚合操作。Graph-Bert 将原始图采样为多个子图，并且只利用attention机制在子图上进行表征学习，而不考虑子图中的边信息。因此Graph-Bert可以解决上面提到的传统GNN具有的性能问题和效率问题。
  + [【关于自训练 + 预训练 = 更好的自然语言理解模型 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/bert_study/SelfTrainingImprovesPreTraining))
    - 论文标题：Self-training Improves Pre-training for Natural Language Understanding
    - 论文地址：https://arxiv.org/abs/2010.02194
    - 动机
      * 问题一: do pre-training and self-training capture the same information, or are they complementary?
      * 问题二: how can we obtain large amounts of unannotated data from specific domains?
    - 方法
      * 问题二解决方法：提出 SentAugment 方法 从 web 上获取有用数据；
      * 问题一解决方法：使用标记的任务数据训练一个 teacher 模型，然后用它对检索到的未标注句子进行标注，并基于这个合成数据集训练最终的模型。
* [【关于 Bert 模型压缩】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/Bert_zip)
  + [【关于 Bert 模型压缩】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/Bert_zip)
    - 阅读理由：Bert 在工程上问题上的改进
    - 动机：
      * 内存占用；
      * 功耗过高；
      * 带来很高的延迟；
      * 限制了 Bert 系列模型在移动和物联网等嵌入式设备上的部署；
    - 介绍：BERT 瘦身来提升速度
    - 模型压缩思路：
      * 低秩因式分解：在输入层和输出层使用嵌入大小远小于原生Bert的嵌入大小，再使用简单的映射矩阵使得输入层的输出或者最后一层隐藏层的输出可以通过映射矩阵输入到第一层的隐藏层或者输出层；
      * 跨层参数共享：隐藏层中的每一层都使用相同的参数，用多种方式共享参数，例如只共享每层的前馈网络参数或者只共享每层的注意力子层参数。默认情况是共享每层的所有参数；
      * 剪枝：剪掉多余的连接、多余的注意力头、甚至LayerDrop[1]直接砍掉一半Transformer层
      * 量化：把FP32改成FP16或者INT8；
      * 蒸馏：用一个学生模型来学习大模型的知识，不仅要学logits，还要学attention score；
    - 优点：BERT 瘦身来提升速度
    - 缺点：
      * 精度的下降
      * 低秩因式分解 and 跨层参数共享 计算量并没有下降；
      * 剪枝会直接降低模型的拟合能力；
      * 量化虽然有提升但也有瓶颈；
      * 蒸馏的不确定性最大，很难预知你的BERT教出来怎样的学生；
  + [【关于 AlBert 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/T5_ALBERT/)
    - 模型压缩方法：低秩因式分解 + 跨层参数共享
    - 模型压缩方法介绍：
      * 低秩因式分解：
        + 动机：Bert的参数量大部分集中于模型的隐藏层架构上，在嵌入层中只有30,000词块，其所占据的参数量只占据整个模型参数量的小部分；
        + 方法：将输入层和输出层的权重矩阵分解为两个更小的参数矩阵；
        + 思路：在输入层和输出层使用嵌入大小远小于原生Bert的嵌入大小，再使用简单的映射矩阵使得输入层的输出或者最后一层隐藏层的输出可以通过映射矩阵输入到第一层的隐藏层或者输出层；
        + 优点：在不显著增加词嵌入大小的情况下能够更容易增加隐藏层大小；
      * 参数共享【跨层参数共享】：
        + 动机：隐藏层 参数 大小 一致；
        + 方法：隐藏层中的每一层都使用相同的参数，用多种方式共享参数，例如只共享每层的前馈网络参数或者只共享每层的注意力子层参数。默认情况是共享每层的所有参数；
        + 优点：防止参数随着网络深度的增加而增大；
    - 其他改进策略：
      * **句子顺序预测损失(SOP)**代替**Bert中的下一句预测损失(NSP)**：
        + 动机：通过实验证明，Bert中的下一句预测损失(NSP) 作用不大；
        + 介绍：用预测两个句子是否连续出现在原文中替换为两个连续的句子是正序或是逆序，用于进一步提高下游任务的表现
    - 优点：参数量上有所降低；
    - 缺点：其加速指标仅展示了训练过程，由于ALBERT的隐藏层架构**采用跨层参数共享策略并未减少训练过程的计算量**，加速效果更多来源于低维的嵌入层；
  + [【关于 FastBERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/FastBERT/)
    - 模型压缩方法：知识蒸馏
    - 模型压缩方法介绍：
      * 样本自适应机制（Sample-wise adaptive mechanism）
        + 思路：

在每层Transformer后都去预测样本标签，如果某样本预测结果的置信度很高，就不用继续计算了，就是自适应调整每个样本的计算量，容易的样本通过一两层就可以预测出来，较难的样本则需要走完全程。

* + - * + 操作：

给每层后面接一个分类器，毕竟分类器比Transformer需要的成本小多了

* + - * 自蒸馏（Self-distillation）
        + 思路：

在预训练和精调阶段都只更新主干参数；

精调完后freeze主干参数，用分支分类器（图中的student）蒸馏主干分类器（图中的teacher）的概率分布

* + - * + 优点：

非蒸馏的结果没有蒸馏要好

不再依赖于标注数据。蒸馏的效果可以通过源源不断的无标签数据来提升

* + [【关于 distilbert】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/distilbert/)
  + [【关于 TinyBert】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/TinyBERT/)
    - 模型压缩方法：知识蒸馏
    - tinybert的创新点：学习了teacher Bert中更多的层数的特征表示；
    - 模型压缩方法介绍：
      * 基于transformer的知识蒸馏模型压缩
        + 学习了teacher Bert中更多的层数的特征表示；
        + 特征表示：

词向量层的输出；

Transformer layer的输出以及注意力矩阵；

预测层输出(仅在微调阶段使用)；

* + - * bert知识蒸馏的过程
        + 左图：整体概括了知识蒸馏的过程

左边：Teacher BERT；

右边：Student TinyBERT

目标：将Teacher BERT学习到的知识迁移到TinyBERT中

* + - * + 右图：描述了知识迁移的细节；

在训练过程中选用Teacher BERT中每一层transformer layer的attention矩阵和输出作为监督信息

* [【关于 Perturbed Masking】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/bert_study/ACL2020_UnsupervisedBert)
  + 论文：Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT
  + 论文链接：https://arxiv.org/pdf/2004.14786.pdf
  + 代码链接：https://github.com/bojone/perturbed\_masking
  + 动机： 通过引入少量的附加参数，probe learns 在监督方式中使用特征表示（例如，上下文嵌入）来 解决特定的语言任务（例如，依赖解析）。这样的probe tasks 的有效性被视为预训练模型编码语言知识的证据。但是，这种评估语言模型的方法会因 probe 本身所学知识量的不确定性而受到破坏。
  + 方法介绍：
    - Perturbed Masking
      * 介绍：parameter-free probing technique
      * 目标：analyze and interpret pre-trained models，测量一个单词xj对预测另一个单词xi的影响，然后从该单词间信息中得出全局语言属性（例如，依赖树）。
  + 思想：整体思想很直接，句法结构，其实本质上描述的是词和词之间的某种关系，如果我们能从BERT当中拿到词和词之间相互“作用”的信息，就能利用一些算法解析出句法结构。

[【关于 信息抽取】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/)

[【关于 实体关系联合抽取】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/)

* [【关于 A Frustratingly Easy Approach for Joint Entity and Relation Extraction】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/AFrustratinglyEasyApproachForJEandRE/) 【强烈推荐】
  + 论文：A Frustratingly Easy Approach for Joint Entity and Relation Extraction
  + 阅读理由：反直觉！陈丹琦用pipeline方式刷新关系抽取SOTA
  + 方法：建立两个 encoders，并独立训练:
    - encoder 1：entity model
      * 方法：建立在 span-level representations 上
    - encoder 2：relation model：只依赖于实体模型作为输入特征
      * 方法：builds on contextual representations specific to a given pair of span
  + 优点：
    - 很简单，但我们发现这种流水线方法非常简单有效；
    - 使用同样的预先训练的编码器，我们的模型在三个标准基准（ACE04，ACE05，SciERC）上优于所有以前的联合模型；
  + 问题讨论：
    - Q1、关系抽取最care什么？
      * 解答：引入实体类别信息会让你的关系模型有提升
    - Q2、共享编码 VS 独立编码 哪家强？
      * 解答：由于两个任务各自是不同的输入形式，并且需要不同的特征去进行实体和关系预测，也就是说：使用单独的编码器确实可以学习更好的特定任务特征。
    - Q3：误差传播不可避免？还是不存在？
      * 解答：并不认为误差传播问题不存在或无法解决，而需要探索更好的解决方案来解决此问题
    - Q4：Effect of Cross-sentence Context
      * 解答：使用跨句上下文可以明显改善实体和关系
* [【关于 实体关系联合抽取】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/%E5%AE%9E%E4%BD%93%E5%85%B3%E7%B3%BB%E8%81%94%E5%90%88%E6%8A%BD%E5%8F%96%E6%80%BB%E7%BB%93.md)
* [Incremental Joint Extraction of Entity Mentions and Relations](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/T2014_joint_extraction/)
* [【关于 Joint NER】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/JointER/)
  + 论文名称：Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy
* [【关于 GraphRel】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/ACL2019_GraphRel/)
  + 论文名称：论文名称：GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction
  + 动机
    - 想要自动提取特征的联合模型
      * 通过堆叠Bi-LSTM语句编码器和GCN (Kipf和Welling, 2017)依赖树编码器来自动学习特征
      * 用以考虑线性和依赖结构
        + 类似于Miwa和Bansal(2016)（一样是堆叠的）

方法

每个句子使用Bi-LSTM进行自动特征学习

提取的隐藏特征由连续实体标记器和最短依赖路径关系分类器共享

问题

然而，在为联合实体识别和关系提取引入共享参数时：

它们仍然必须将标记者预测的实体提及通过管道连接起来

形成关系分类器的提及对

* + - * 考虑重叠关系
      * 如何考虑关系之间的相互作用
        + 2nd-phase relation-weighted GCN
        + 重叠关系(常见）

情况

两个三元组的实体对重合

两个三元组都有某个实体mention

推断

困难（对联合模型尤其困难，因为连实体都还不知道）

* + - 方法：
      * 学习特征
        + 通过堆叠Bi-LSTM语句编码器和GCN (Kipf和Welling, 2017)依赖树编码器来自动学习特征
      * 第一阶段的预测
        + GraphRel标记实体提及词，预测连接提及词的关系三元组
        + 用关系权重的边建立一个新的全连接图（中间图）
        + 指导：关系损失和实体损失
      * 第二阶段的GCN
        + 通过对这个中间图的操作
        + 考虑实体之间的交互作用和可能重叠的关系
        + 对每条边进行最终分类
        + 在第二阶段，基于第一阶段预测的关系，我们为每个关系构建完整的关系图，并在每个图上应用GCN来整合每个关系的信息，进一步考虑实体与关系之间的相互作用。
* [【关于 HBT】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/ERE_study/T20ACL_HBT_su/)
  + 论文名称：A Novel Hierarchical Binary Tagging Framework for Relational Triple Extraction
  + 动机：
    - pipeline approach
      * 思路
        + 识别句子中的所有实体；
        + 对每个实体对执行关系分类。
      * 问题
        + 由于早期阶段的错误无法在后期阶段进行纠正，因此这种方法容易遭受错误传播问题的困扰。
    - feature-based models and neural network-based models
      * 思路
        + 通过用学习表示替换人工构建的特征，基于神经网络的模型在三重提取任务中取得了相当大的成功。
      * 问题
        + 大多数现有方法无法正确处理句子包含多个相互重叠的关系三元组的情况。
    - 基于Seq2Seq模型 and GCN
      * 思路：提出了具有复制机制以提取三元组的序列到序列（Seq2Seq）模型。 他们基于Seq2Seq模型，进一步研究了提取顺序的影响，并通过强化学习获得了很大的改进。
      * 问题：它们都将关系视为要分配给实体对的离散标签。 这种表述使关系分类成为硬机器学习问题。 首先，班级分布高度不平衡。 在所有提取的实体对中，大多数都不形成有效关系，从而产生了太多的否定实例。 其次，当同一实体参与多个有效关系（重叠三元组）时，分类器可能会感到困惑。 没有足够的训练示例，分类器就很难说出实体参与的关系。结果，提取的三元组通常是不完整且不准确的。
  + 方法：
    - 首先，我们确定句子中所有可能的 subjects；
    - 然后针对每个subjects，我们应用特定于关系的标记器来同时识别所有可能的 relations 和相应的 objects。

[【关于 命名实体识别】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/)：

* [【关于 LEX-BERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ICLR2021_LEX_BERT/)【强烈推荐】
  + 推荐理由：在 query 中 引入 标签信息的方法，秒杀 Flat NER，登上 2021 年 Chinese NER SOTA。
  + 论文名称：《Lex-BERT: Enhancing BERT based NER with lexicons》
  + 动机：尽管它在NER任务中的表现令人印象深刻，但最近已经证明，添加词汇信息可以显著提高下游性能。然而，没有任何工作在不引入额外结构的情况下将单词信息纳入BERT。在我们的工作中，我们提出了词法BERT（lex-bert），这是一种在基于BERT的NER模型中更方便的词汇借用方法
  + 方法：
    - LEX-BERT V1：Lex BERT的第一个版本通过在单词的左右两侧插入特殊标记来识别句子中单词的 span。特殊标记不仅可以标记单词的起始位置和结束位置，还可以为句子提供实体类型信息
    - LEX-BERT V2：对于在句子中加宽的单词，我们没有在句子中单词的周围插入起始和结束标记，而是在句子的末尾附加一个标记[x]。请注意，我们将标记的位置嵌入与单词的起始标记绑定
* [【关于 Biaffine Ner 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2020_NERasDependencyParsing/)
  + 动机：NER 研究 关注于 扁平化NER，而忽略了 实体嵌套问题；
  + 方法： 在本文中，我们使用基于图的依存关系解析中的思想，以通过 biaffine model 为模型提供全局的输入视图。 biaffine model 对句子中的开始标记和结束标记对进行评分，我们使用该标记来探索所有跨度，以便该模型能够准确地预测命名实体。
  + 工作介绍：在这项工作中，我们将NER重新确定为开始和结束索引的任务，并为这些对定义的范围分配类别。我们的系统在多层BiLSTM之上使用biaffine模型，将分数分配给句子中所有可能的跨度。此后，我们不用构建依赖关系树，而是根据候选树的分数对它们进行排序，然后返回符合 Flat 或 Nested NER约束的排名最高的树 span；
  + 实验结果：我们根据三个嵌套的NER基准（ACE 2004，ACE 2005，GENIA）和五个扁平的NER语料库（CONLL 2002（荷兰语，西班牙语），CONLL 2003（英语，德语）和ONTONOTES）对系统进行了评估。结果表明，我们的系统在所有三个嵌套的NER语料库和所有五个平坦的NER语料库上均取得了SoTA结果，与以前的SoTA相比，实际收益高达2.2％的绝对百分比。
* [【关于 NER trick】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/NER_study/NERtrick.md)
* [【关于TENER】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2019/ACL2019_TENER/)
  + 论文名称：TENER: Adapting Transformer Encoder for Name Entity Recognition
  + 动机：
    - 1. Transformer 能够解决长距离依赖问题；
    - 2. Transformer 能够并行化；
    - 3. 然而，Transformer 在 NER 任务上面效果不好。
  + 方法：
    - 第一是经验发现。 引入：相对位置编码
    - 第二是经验发现。 香草变压器的注意力分布是缩放且平滑的。 但是对于NER，因为并非所有单词都需要参加，所以很少注意是合适的。 给定一个当前单词，只需几个上下文单词就足以判断其标签。 平稳的注意力可能包括一些嘈杂的信息。 因此，我们放弃了点生产注意力的比例因子，而使用了无比例且敏锐的注意力。
* [【关于DynamicArchitecture】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/)
  + 介绍：Dynamic Architecture范式通常需要设计相应结构以融入词汇信息。
  + 论文：
    - [【关于 LatticeLSTM 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/1_ACL2018_LatticeLSTM/)
      * 想法：在 char-based 的 LSTM 中引入词汇信息
      * 做法：
        + 根据大量语料生成词典；
        + 若当前字符与前面的字符无法组成词典中词汇，则按 LSTM 的方法更新记忆状态；
        + 若当前字符与前面的字符组成词典中词汇，从最新词汇中提取信息，联合更新记忆状态；
      * 存在问题：
        + 计算性能低下，导致其**不能充分利用GPU进行并行化**。究其原因主要是每个字符之间的增加word cell（看作节点）数目不一致；
        + 信息损失：

1）每个字符只能获取以它为结尾的词汇信息，对于其之前的词汇信息也没有持续记忆。如对于「大」，并无法获得‘inside’的「长江大桥」信息。

2）由于RNN特性，采取BiLSTM时其前向和后向的词汇信息不能共享，导致 Lattice LSTM **无法有效处理词汇信息冲突问题**

* + - * + 可迁移性差：只适配于LSTM，不具备向其他网络迁移的特性。
    - [【关于 LR-CNN 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/2_IJCAI2019_LR_CNN/)
      * 动机
        + 词信息引入问题；

lattice LSTM 问题：

基于 RNN 结构方法不能充分利用 GPU 并行计算资源；

针对句子中字符计算；

针对匹配词典中潜在词

很难处理被合并到词典中的潜在单词之间的冲突：

一个字符可能对应词典中多个潜在词，误导模型

方法：

* + - * + Lexicon-Based CNNs：采取CNN对字符特征进行编码，感受野大小为2提取bi-gram特征，堆叠多层获得multi-gram信息；同时采取注意力机制融入词汇信息（word embed）；
        + Refining Networks with Lexicon Rethinking：由于上述提到的词汇信息冲突问题，LR-CNN采取rethinking机制增加feedback layer来调整词汇信息的权值：具体地，将高层特征作为输入通过注意力模块调节每一层词汇特征分布，利用这种方式来利用高级语义来完善嵌入单词的权重并解决潜在单词之间的冲突。
    - [【关于 CGN 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/3_EMNLP2019_CGN/)
      * 动机
        + 中文命名实体识别中，词边界 问题；
        + 如何 引入 词边界信息：

pipeline：CWS -> NER

问题：误差传递

CWS 和 NER 联合学习

问题：标注 CWS 数据

利用 词典 自动构建

优点：比 CWS 标注数据 更容易获得

问题：

第一个挑战是整合自我匹配的词汇词；

举例：“北京机场” (Beijing Airport) and “机场” (Airport) are the self-matched words of the character “机” (airplane)

第二个挑战是直接整合最接近的上下文词汇词；

举例：by directly using the semantic knowledge of the nearest contextual words “离开” (leave), an “I-PER” tag can be predicted instead of an “I-ORG” tag, since “?希尔顿” (Hilton Hotels) cannot be taken as the subject of the verb “离开”

* + - * + 论文思路：

character-based Collaborative Graph：

encoding layer：

句子信息：

s1：将 char 表示为 embedding;

s2：利用 biLSTM 捕获 上下文信息

lexical words 信息：

s1：将 lexical word 表示为 embedding;

合并 contextual representation 和 word embeddings

a graph layer：

Containing graph (C-graph):

思路：字与字之间无连接，词与其inside的字之间有连接；

目的：帮助 字符 捕获 self-matched lexical words 的边界和语义信息

Transition graph (T-graph):

思路：相邻字符相连接，词与其前后字符连接；

目的：帮助 字符 捕获 相邻 上下文 lexical 词 的 语义信息

Lattice graph (L-graph):

思路：通相邻字符相连接，词与其开始结束字符相连；

目的：融合 lexical knowledge

GAT:

操作：针对三种图，使用Graph Attention Network(GAN)来进行编码。最终每个图的输出

> 其中 图片 为第k个图的GAN表示，因为是基于字符级的序列标注，所以解码时只关注字符，因此从矩阵中取出前n行作为最终的图编码层的输出。

a fusion layer：

目的：融合 三种 graphs 中不同 的 lexical 知识

a decoding layer:

操作：利用 CRF 解码

* + - [【关于 LGN 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/4_EMNLP2019_LGN/)
      * 动机：
        + 在 char-base Chinese NER 中，同一个字符可能属于多个 lexicon word，存在 overlapping ambiguity 问题

举例(如下图)

字符流 和 流经；

* + - * + Lattice LSTM这种RNN结构仅仅依靠前一步的信息输入，而不是利用全局信息

举例

字符 度 信息，从而造成标注冲突问题

* + - * + Ma等人于2014年提出，想解决overlapping across strings的问题，需要引入「整个句子中的上下文」以及「来自高层的信息」；然而，现有的基于RNN的序列模型，不能让字符收到序列方向上 remain characters 的信息；
      * 方法：
        + Graph Construction and Aggregation
        + Graph Construction
        + Local Aggregation
        + Global Aggregation
        + Recurrent-based Update Module
    - [【关于 FLAT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/DynamicArchitecture/5_ACL2020_FLAT/)
      * 动机
        + 方法一：设计一个动态框架，能够兼容词汇输入；

代表模型：

Lattice LSTM：利用额外的单词单元编码可能的单词，并利用注意力机制融合每个位置的变量节点

LR-CNN：采用不同窗口大小的卷积核来编码 潜在词

问题：

RNN 和 CNN 难以解决长距离依赖问题，它对于 NER 是有用的，例如： coreference（共指）

无法充分利用 GPU 的并行计算能力

* + - * + 方法二：将 Lattice 转化到图中并使用 GNN 进行编码：

代表模型

Lexicon-based GN(LGN)

Collaborative GN(CGN)

问题

虽然顺序结构对于NER仍然很重要，并且 Graph 是一般的对应物，但它们之间的差距不可忽略;

需要使用 LSTM 作为底层编码器，带有顺序感性偏置，使模型变得复杂。

* + - * 方法：将Lattice结构展平，将其从一个有向无环图展平为一个平面的Flat-Lattice Transformer结构，由多个span构成：每个字符的head和tail是相同的，每个词汇的head和tail是skipped的。
* [【关于 ACL 2019 中的NER】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2019/)
  + [named entity recognition using positive-unlabeled learning](https://github.com/km1994/nlp_paper_study/tree/master/NER_study/ACL2019/JointER/)
  + [【关于 GraphRel】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2019/ACL2019_NERusingPositive-unlabeledLearning/)
    - 论文名称：GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction
  + [Fine-Grained Entity Typing in Hyperbolic Space（在双曲空间中打字的细粒度实体）](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2019/Fine-GrainedEntityTypinginHyperbolicSpace/)
  + [【关于 TENER】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/ACL2019/ACL2019_TENER/)
    - 论文名称：TENER: Adapting Transformer Encoder for Name Entity Recognition
* [【关于 EMNLP 2019 中的NER】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/EMNLP2019/)
  + [CrossWeigh从不完善的注释中训练命名实体标注器](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/EMNLP2019/CrossWeigh%E4%BB%8E%E4%B8%8D%E5%AE%8C%E5%96%84%E7%9A%84%E6%B3%A8%E9%87%8A%E4%B8%AD%E8%AE%AD%E7%BB%83%E5%91%BD%E5%90%8D%E5%AE%9E%E4%BD%93%E6%A0%87%E6%B3%A8%E5%99%A8/)
  + [利用词汇知识通过协同图网络进行中文命名实体识别](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/EMNLP2019/%E5%88%A9%E7%94%A8%E8%AF%8D%E6%B1%87%E7%9F%A5%E8%AF%86%E9%80%9A%E8%BF%87%E5%8D%8F%E5%90%8C%E5%9B%BE%E7%BD%91%E7%BB%9C%E8%BF%9B%E8%A1%8C%E4%B8%AD%E6%96%87%E5%91%BD%E5%90%8D%E5%AE%9E%E4%BD%93%E8%AF%86%E5%88%AB/)
  + [一点注释对引导低资源命名实体识别器有很多好处](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NER_study/EMNLP2019/%E4%B8%80%E7%82%B9%E6%B3%A8%E9%87%8A%E5%AF%B9%E5%BC%95%E5%AF%BC%E4%BD%8E%E8%B5%84%E6%BA%90%E5%91%BD%E5%90%8D%E5%AE%9E%E4%BD%93%E8%AF%86%E5%88%AB%E5%99%A8%E6%9C%89%E5%BE%88%E5%A4%9A%E5%A5%BD%E5%A4%84/)

[【关于 关系抽取】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/)：

* [End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures【2016】](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T2016_LSTM_Tree/)
* [【关于 ERNIE】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/NRE_paper_study/ERNIE/)
* [【关于 GraphRel】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/GraphRel/)
* [【关于 R\_BERT】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/R_BERT)
* [【关于 Task 1：全监督学习】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T1_FullySupervisedLearning/)
  + [Relation Classification via Convolutional Deep Neural Network](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T1_FullySupervisedLearning/T1_Relation_Classification_via_CDNN/)
  + [Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T1_FullySupervisedLearning/T2_Attention-Based_BiLSTM_for_RC/)
  + [Relation Classification via Attention Model](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/information_extraction/NRE_paper_study/T1_FullySupervisedLearning/T3_RC_via_attention_model_new/)
* [【关于 Task 2：远程监督学习】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T2_DistantSupervisedLearning/)
  + [Relation Classification via Convolutional Deep Neural Network](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T2_DistantSupervisedLearning/T1_Piecewise_Convolutional_Neural_Networks/)
  + [NRE\_with\_Selective\_Attention\_over\_Instances](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/NRE_paper_study/T2_DistantSupervisedLearning/T2_NRE_with_Selective_Attention_over_Instances/)

[【关于 文档级别关系抽取】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/Doc-level_Relation_Extraction/)

* [【关于 Double Graph Based Reasoning for Document-level Relation Extraction】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/information_extraction/Doc-level_Relation_Extraction/DoubleGraphBasedReasoningforDocumentlevelRelationExtraction/)

[【关于 知识图谱 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/)

【[实体链指篇](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/entity_linking/)】

* [【关于 Low-resource Cross-lingual Entity Linking】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/entity_linking/LowResourceCrossLingualEntityLinking/)
  + 论文名称：Design Challenges in Low-resource Cross-lingual Entity Linking
  + 论文地址：https://arxiv.org/pdf/2005.00692.pdf
  + 来源：EMNLP 2020
* [【关于 GENRE】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/entity_linking/GENRE_ICLR21/)
  + 论文名称：AUTOREGRESSIVE ENTITY RETRIEVAL
  + 论文地址：https://openreview.net/pdf?id=5k8F6UU39V
  + 来源：EMNLP 2020

【[【关于 实体消歧 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/EntityDisambiguation/)】

* [【关于 DeepType 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/EntityDisambiguation/DeepType/)
  + 论文：DeepType: Multilingual Entity Linking by Neural Type System Evolution
  + 论文地址：https://arxiv.org/abs/1802.01021
  + github：https://github.com/openai/deeptype

【[【关于KGQA 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/KGQA/)】

* [【关于KGQA 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/KGQA/)
  + 一、基于词典和规则的方法
  + 二、基于信息抽取的方法
* [【关于 Multi-hopComplexKBQA 】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/KGQA/ACL20_Multi-hopComplexKBQA/)
  + 论文：Lan Y, Jiang J. Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 969-974.
  + 会议：ACL2020
  + 链接：https://www.aclweb.org/anthology/2020.acl-main.91/
  + 代码：https://github.com/lanyunshi/Multi-hopComplexKBQA

【[【关于Neo4j 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/KG_study/neo4j/)】

* 一、数据导入

[【关于 细粒度情感分析】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/ABSC_study/)

* [【关于 LCF】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/ABSC_study/LCF/)
  + 论文名称：A Local Context Focus Mechanism for Aspect-Based Sentiment Classiﬁcation
  + 论文动机：没有考虑情感极性和局部上下文间关系
    - LCF:利用自注意力机制同时捕获局部上下文特征和全局上下文特征，以推断 targeted aspect 的情感极性
    - SRD:评估上下文词与 aspect 间的独立性，SRD对于弄清局部上下文具有重要意义，并且SRD阈值中的上下文单词的特征将得到保留和重点关注。
    - CDM 和 CDW 层：强化 LCF，使其对 特殊 aspest 的局部上下文提供 更多 注意力。CDM层通过掩盖语义相对较少的上下文词的输出表示，将重点放在局部上下文上。 CDW 层根据 SRD 削弱语义相对较少的上下文词的特征；

[【关于 主动学习】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/active_learn_study/)

* [【关于 Proactive Learning for Named Entity Recognition（命名实体识别的主动学习）】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/active_learn_study/ProactiveLearningforNamedEntityRecognition/)

[【关于 对抗训练】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/adversarial_training_study/)

* [【关于 生成对抗网络 GAN 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DL_algorithm/adversarial_training_study/)
* [【关于 FreeLB 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/adversarial_training_study/FREELB/)
  + 论文名称: FreeLB: Enhanced Adversarial Training for Language Understanding 加强语言理解的对抗性训练
  + 动机：对抗训练使保留标签的输入扰动的最大风险最小，对于提高语言模型的泛化能力是有效的。
  + 方法：提出了一种新的对抗性训练算法—— freeb，它通过在字嵌入中添加对抗性的干扰，最小化输入样本周围不同区域内的对抗性风险，从而提高嵌入空间的鲁棒性和不变性。

[【关于 GCN in NLP 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/GNN/GCN2NLP/)：

* [【关于 GCN in NLP 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/GNN/GCN2NLP/readme.md)
  + 构图方法：
    - 句法依赖树；
    - TF-IDF;
    - PMI;
    - 序列关系；
    - 词典

[【关于 文本预处理】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/NLP_tools/pre_study/samplingStudy/)

* [【关于 过采样】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/NLP_tools/pre_study/samplingStudy/samplingStudy)

[【关于问答系统】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/)

* [【关于 FAQ Trick】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/FAQ/FAQ_trick/)
* [【关于 文本匹配和多轮检索】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/%E6%96%87%E6%9C%AC%E5%8C%B9%E9%85%8D%E5%92%8C%E5%A4%9A%E8%BD%AE%E6%A3%80%E7%B4%A2.xmind)
* [【关于 FAQ】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/FAQ/)
  + [【关于 LCNQA】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/LCNQA/)
    - 论文名称：Lattice CNNs for Matching Based Chinese Question Answering
  + [LSTM-based Deep Learning Models for Non-factoid Answer Selection](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/T1_LSTM-based_for_Non-factoid_Answer_Selection/)
  + [【关于 Denoising Distantly Supervised ODQA】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/T4_DenoisingDistantlySupervisedODQA/)
    - 论文名称：Denoising Distantly Supervised Open-Domain Question Answering
  + [FAQ retrieval using query-question similarity and BERT-based query-answer relevance](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/ACM2019_faq_bert-based_query-answer_relevance/)
  + [【DC-BERT】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/SIGIR2020_DCBert/)
    - 论文名称：DC-BERT : DECOUPLING QUESTION AND DOCUMENT FOR EFFICIENT CONTEXTUAL ENCODING
* [【关于 KBFAQ】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/KBFAQ/)
* [【关于 MulFAQ】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/mulFAQ/)
  + [【关于 MSN】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/mulFAQ/MSN_mulQA/)
    - 论文名称：Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based chatbots
    - 论文地址：https://www.aclweb.org/anthology/D19-1011.pdf
    - 论文项目：https://github.com/chunyuanY/Dialogue
    - 动机：
      * 1. 上下文拼接问题：将候选回复与上下文utterance在多粒度级别进行匹配，这种方式忽略了使用过多的上下文信息带来副作用。
      * 2. 根据直接，一般来说距离回复越近的utterance，越能够反应最终轮对话的意图。所以，我们首先使用最后一个utterance作为key去选择word级别和sentence级别上相关的上下文回复。然而，我们发现**许多样本中最后一个utterance都是一些短句并且很多都是无效信息**（比如good, ok）
    - 方法：提出一种多跳选择网络（multi-hop selector network, MSN）
      * s1 ：采用 多跳选择器从 上下文集 中 选取最相关的上下文 utterances，并生成 k 个 不同的上下文；
      * s2 : 融合 k 个 上下文 utterance ，并与候选回复做匹配；
      * s3 : 匹配截取，采用 CNN 提取匹配特征，并 用 GRU 学习 utterance 间的临时关系；

[【关于 文本摘要】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/summarization_study/)

* [【关于 Bertsum】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/summarization_study/EMNLP2019_bertsum/) **【推荐阅读】**
  + 论文名称：Fine-tune BERT for Extractive Summarization
  + 会议：EMNLP2019
  + Bert 在抽取式文本摘要中的应用
    - 问题：
      * 如何获得每个句子向量？
      * 如何将向量用于二分类问题？
      * 如何判断每个句子的去留？
  + 思路：定义文档 图片,图片 表示 文档中的第图片个句子，Extractive summarization 定义为 给每个图片分配一个标签图片，用于判断该句子是否包含于 摘要 中。
    - 方法介绍
      * Extractive Summarization with BERT
        + 动机：

Bert 作为 一个 masked-language model，输出向量基于标记而不是句子；

Bert 只包含 两个标签（sentence A or sentence B），而不是多个句子；

* + - * + 方法

Encoding Multiple Sentences：在每个句子之前插入一个CLS token

Interval Segment Embeddings

我们使用区间段嵌入来区分文档中的多个句子。

例如：对于$sent\_1，sent\_2，sent\_3，sent\_4，sent\_5

图片 由向量 图片 表示，图片是来自顶部BERT层的第图片个[CLS]符号的向量。

* [【关于Pointer-Generator Networks 指针网络】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/summarization_study/ACL2017_Pointer_Generator_Networks/)
  + 论文名称：Get To The Point: Summarization with Pointer-Generator Networks
  + 会议：ACL2017
  + 动机：
    - 文本摘要类别
      * extractive 抽取式
        + 方式：直接从原文抽取一些段落
        + 优点：简单
        + 问题：无法生成高质量的摘要，因为不具备一些复杂的摘要能力(如释义(paraphasing), 概括(generalization), 与现实世界知识的融合(incorporation of real-world knowledge))
      * abstractive 生成式
        + 方式：根据长文本 生成 摘要
        + 代表：seq2sq架构
        + 问题：

难以准确复述原文细节；

无法处理原文中的未登录词(OOV)；

在生成的摘要中存在一些重复的部分；

* + 方法：
    - 编码器（encoder）
      * 方式：BiLSTM
      * 作用：将文章中每个词的词向量编码为 隐状态 ht
    - 解码器（decoder）
      * 方式：单向 LSTM
      * 作用：每一时刻 t，将上一时刻 生成 的 词的词向量作为输入，得到 Decoder Hidden State st，该状态被用于计算attention分布和词的预测
    - Attention
      * 作用：每个时间步t,考虑当前序列的注意力分布，注意力分布用于生成编码器隐藏状态的加权总和，转化为上下文向量，与解码器t时刻的隐状态进行concatenated然后喂到两个线性层来计算词汇分布P（一个固定维度的向量，每个维度代表被预测词的概率，取argmax就能得到预测的单词）。
      * 目的：告诉模型在当前步的预测过程中，原文中的哪些词更重要

[【关于 文本匹配】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/)

* [【关于 语义相似度匹配任务中的 BERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/bert_similairity/) **【推荐阅读】**
  + 阅读理由：BERT 在 语义相似度匹配任务 中的应用，可以由很多种方式，然而，你真的了解这些方式的区别和优缺点么？
  + 动机：BERT 在 语义相似度匹配任务 中的应用，可以常用 Sentence Pair Classification Task：使用 [CLS]、cosine similairity、sentence/word embedding、siamese network 方法，那么哪种是最佳的方式呢？你是否考虑过呢?
* [【关于 MPCNN】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/Multi-PerspectiveSentenceSimilarityModelingwithCNN/)
  + 论文：Multi-Perspective Sentence Similarity Modeling with Convolution Neural Networks
* [【关于 RE2】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/Multi-RE2_study/)
  + 论文：Simple and Effective Text Matching with Richer Alignment Features
  + 动机： 可以使用多个序列间比对层构建更强大的模型。 代替基于单个对准过程的比较结果进行预测，具有多个对准层的堆叠模型将保持其中间状态并逐渐完善其预测。**但是，由于底层特征的传播效率低下和梯度消失，这些更深的体系结构更难训练。**
  + 介绍：一种快速强大的神经体系结构，具有用于通用文本匹配的多个对齐过程。 我们对以前文献中介绍的文本匹配方法中许多慢速组件的必要性提出了质疑，包括复杂的多向对齐机制，对齐结果的大量提炼，外部句法特征或当模型深入时用于连接堆叠块的密集连接。 这些设计选择会极大地减慢模型的速度，并且可以用重量更轻且效果相同的模型代替。 同时，我们重点介绍了有效文本匹配模型的三个关键组成部分。 这些组件（名称为RE2代表）是以前的对齐特征（残差矢量），原始点向特征（嵌入矢量）和上下文特征（编码矢量）。 其余组件可能尽可能简单，以保持模型快速，同时仍能产生出色的性能。
* [【关于 DSSM】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/cikm2013_DSSM/)
  + 论文：Deep Structured Semantic Model
  + 论文会议：CIKM2013
  + 问题：语义相似度问题
    - 字面匹配体现
      * 召回：在召回时，传统的文本相似性如 BM25，无法有效发现语义类 Query-Doc 结果对，如"从北京到上海的机票"与"携程网"的相似性、"快递软件"与"菜鸟裹裹"的相似性
      * 排序：在排序时，一些细微的语言变化往往带来巨大的语义变化，如"小宝宝生病怎么办"和"狗宝宝生病怎么办"、"深度学习"和"学习深度"；
    - 使用 LSA 类模型进行语义匹配，但是效果不好
  + 思路：
    - 利用 表示层 将 Query 和 Title 表达为低维语义向量；
    - 通过 cosine 距离来计算两个语义向量的距离，最终训练出语义相似度模型。
  + 优点
    - 减少切词的依赖：解决了LSA、LDA、Autoencoder等方法存在的一个最大的问题，因为在英文单词中，词的数量可能是没有限制，但是字母 n-gram 的数量通常是有限的
    - 基于词的特征表示比较难处理新词，字母的 n-gram可以有效表示，鲁棒性较强；
    - 传统的输入层是用 Embedding 的方式（如 Word2Vec 的词向量）或者主题模型的方式（如 LDA 的主题向量）来直接做词的映射，再把各个词的向量累加或者拼接起来，由于 Word2Vec 和 LDA 都是无监督的训练，这样会给整个模型引入误差，DSSM 采用统一的有监督训练，不需要在中间过程做无监督模型的映射，因此精准度会比较高；
    - 省去了人工的特征工程；
  + 缺点
    - word hashing可能造成冲突
    - DSSM采用了词袋模型，损失了上下文信息
    - 在排序中，搜索引擎的排序由多种因素决定，由于用户点击时doc的排名越靠前，点击的概率就越大，如果仅仅用点击来判断是否为正负样本，噪声比较大，难以收敛
* [【关于 ABCNN 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/TACL2016_ABCNN/)
  + 论文：ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs
  + 会议：TACL 2016
  + 论文方法：采用了CNN的结构来提取特征，并用attention机制进行进一步的特征处理，作者一共提出了三种attention的建模方法
* [【关于 ESIM 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/TACL2017_ESIM/)
  + 论文：Enhanced LSTM for Natural Language Inference
  + 会议：TACL2017
  + 自然语言推理（NLI: natural language inference）问题：
    - 即判断能否从一个前提p中推导出假设h
    - 简单来说，就是判断给定两个句子的三种关系：蕴含、矛盾或无关
  + 论文方法：
    - 模型结构图分为左右两边：
    - 左侧就是 ESIM，
    - 右侧是基于句法树的 tree-LSTM，两者合在一起交 HIM (Hybrid Inference Model)。
    - 整个模型从下往上看，分为三部分：
      * input encoding；
      * local inference modeling；
      * inference composition；
      * Prediction
* [【关于 BiMPM 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/IJCAI2017_BiMPM/)
  + 论文：Bilateral multi-perspective matching for natural language sentences
  + 会议：IJCAI2017
  + 方法：
    - Word Representation Layer:其中词表示层使用预训练的Glove或Word2Vec词向量表示, 论文中还将每个单词中的字符喂给一个LSTM得到字符级别的字嵌入表示, 文中使用两者构造了一个dd维的词向量表示, 于是两个句子可以分别表示为 P:p1,⋯,pm.
    - Context Representation Layer: 上下文表示层, 使用相同的双向LSTM来对两个句子进行编码. 分别得到两个句子每个时间步的输出.
    - Matching layer: 对两个句子PP和QQ从两个方向进行匹配, 其中⊗⊗表示某个句子的某个时间步的输出对另一个句子所有时间步的输出进行匹配的结果. 最终匹配的结果还是代表两个句子的匹配向量序列.
    - Aggregation Layer: 使用另一个双向LSTM模型, 将两个匹配向量序列两个方向的最后一个时间步的表示(共4个)进行拼接, 得到两个句子的聚合表示.
* Prediction Layer: 对拼接后的表示, 使用全连接层, 再进行softmax得到最终每个标签的概率.
* [【关于 DIIN 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/text_match_study/T2017_DIIN/)
  + 论文：Densely Interactive Inference Network
  + 会议：TACL2017
  + 模型主要包括五层：嵌入层（Embedding Layer）、编码层（Encoding Layer）、交互层（Interaction Layer ）、特征提取层（Feature Extraction Layer）和输出层（Output Layer）
* [【关于 DC-BERT】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/SIGIR2020_DCBert/)
  + 论文名称：DC-BERT : DECOUPLING QUESTION AND DOCUMENT FOR EFFICIENT CONTEXTUAL ENCODING
  + 阅读理由：Bert 在 QA 上面的应用
  + 动机：Bert 无法处理传入问题的高吞吐量，每个问题都有大量检索到的文档；
  + 论文方法：具有双重BERT模型的解耦上下文编码框架：
    - 一个在线BERT，仅对问题进行一次编码；
    - 一个正式的BERT，对所有文档进行预编码并缓存其编码；
* [【关于 tBERT 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/QA_study/SIGIR2020_DCBert/)
  + 论文：tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection
  + 会议：ACL2020
  + 论文地址：https://www.aclweb.org/anthology/2020.acl-main.630/
  + 论文代码：https://github.com/wuningxi/tBERT
  + 动机：未存在将主题模型和BERT结合的方法。 语义相似度检测是自然语言的一项基本任务理解。添加主题信息对于以前的特征工程语义相似性模型和神经网络模型都是有用的其他任务。在那里目前还没有标准的方法将主题与预先训练的内容表示结合起来比如 BERT。
  + 方法：我们提出了一种新颖的基于主题的基于BERT的语义相似度检测体系结构，并证明了我们的模型在不同的英语语言数据集上的性能优于强神经基线。我们发现在BERT中添加主题特别有助于解决特定领域的情况。

[【关于 机器翻译】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/MachineTranslation/)

* [Neural Machine Translation of Rare Words with Subword Units 论文学习](https://github.com/km1994/nlp_paper_study/tree/master/MachineTranslation/NeuralMachineTranslationOfRareWordsWithSubwordUnits/)

[【关于 文本生成】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/TextGeneration/)

* [【关于 SLCVAE 安装 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/TextGeneration/SLCVAE/)

[【关于 实体消歧】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/EntityDisambiguation/)

* [【关于 DeepType】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/EntityDisambiguation/DeepType/)
  + 阅读理由：这篇论文属于 实体消歧 领域 比较经典的论文，而且调研了很多相关竞赛，很多 Top 大佬 都会用到它，所以想学习一下！【目前还没看完！！！】

[【关于 对话系统】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/)

1. [【关于 Domain/Intent Classification 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/IntentClassification/)
2. [【关于 槽位填充 (Slot Filling)】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/SlotFilling/)
3. [【关于 上下文LU】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/contextLU/)
4. [【关于 自然语言生成NLG 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/NLG/)
5. [【关于 DSTC 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/DSTC/)
6. [【关于 E2E 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/E2E/)
   1. [【关于 TC\_Bot(End-to-End Task-Completion Neural Dialogue Systems) 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/E2E/TC_Bot/)

[【关于 Rasa 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/)

1. [【关于 rasa 安装 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E5%AE%89%E8%A3%85%E6%89%8B%E5%86%8C.md)
2. [【关于 rasa 基本架构 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E5%9F%BA%E6%9C%AC%E6%A1%86%E6%9E%B6_%E8%A7%86%E9%A2%91%E8%AE%B2%E8%A7%A3.md)
3. [【关于 rasa中文对话系统】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E4%B8%AD%E6%96%87%E5%AF%B9%E8%AF%9D%E7%B3%BB%E7%BB%9F.md)
4. [【关于 rasa中文对话系统构建】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E4%B8%AD%E6%96%87%E5%AF%B9%E8%AF%9D%E7%B3%BB%E7%BB%9F%E6%9E%84%E5%BB%BA.md)
5. [【关于 rasa->NLU 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E7%B3%BB%E5%88%97/rasa_nlu.md)
6. [【关于 rasa -> Core -> FormAction 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa_core_FormAction/rasa_nlu.md)
7. [【关于 rasa -> Core -> Stories 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa%E7%B3%BB%E5%88%97/rasa_core_Stories.md)
8. [【关于 rasa -> Core -> Action 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/DialogueSystem_study/rasa/rasa_core_FormAction/rasa_core_Action.md)

[【关于 半监督学习】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/Unsupervised/)

* [Unsupervised Data Augmentation (UDA)](https://github.com/km1994/nlp_paper_study/tree/master/Unsupervised/UDA/)
  + [【关于 UDA】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/Unsupervised/UDA/)
    - 阅读理由：UDA（Unsupervised Data Augmentation 无监督数据增强）是Google在2019年提出的半监督学习算法。该算法超越了所有现有的半监督学习方法，并实现了仅使用极少量标记样本即可达到使用大量标记样本训练集的精度。
    - 动机： 深度学习的模型训练通常依赖大量的标签数据，在只有少量数据上通常表现不好;
    - 思路：提出了一种基于无监督数据的数据增强方式UDA（Unsupervised Data Augmentation）。UDA方法生成无监督数据与原始无监督数据具备分布的一致性，而以前的方法通常只是应用高斯噪声和dropout噪声（无法保证一致性）。UDA方法利用了一种目前为止最优的方法生成更加“真实”的数据。
    - 优点：使用这种数据增强方法，在极少量数据集上，六种语言任务和三种视觉任务都得到了明显的提升。
* [【关于 “脏数据”处理】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/Unsupervised/noisy_label_learning/)
  + 一、动机
    - 1.1 何为“脏数据”？
    - 1.2 “脏数据” 会带来什么后果？
      * 二、“脏数据” 处理篇
    - 2.1 “脏数据” 怎么处理呢？
    - 2.2 置信学习方法篇
      * 2.2.1 什么是 置信学习方法？
      * 2.2.2 置信学习方法 优点？
      * 2.2.3 置信学习方法 怎么做？
      * 2.2.4 置信学习方法 怎么用？有什么开源框架？
      * 2.2.5 置信学习方法 的工作原理？

[【关于 NLP分类任务】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/classifier_study/)

* [【关于 NLP分类任务】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/classifier_study/)
* [【关于 Knowledge in TextCNN】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/classifier_study/KG_classifier/)
  + 论文： Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification
  + github：https://zhuanlan.zhihu.com/p/183852900
  + 介绍：文本分类是NLP应用程序中的一项基本任务。 现有的大多数工作都依靠显式或隐式文本表示来解决此问题。
  + 问题：虽然这些技术对句子很有效，但由于其简短和稀疏，因此无法轻松地应用于短文本。
  + 方法：提出了一个基于卷积神经网络的框架，该框架结合了短文本的显式和隐式表示形式进行分类
    - 首先使用大型分类学知识库将短文本概念化为一组相关概念。
    - 然后，通过在预训练的单词向量之上合并单词和相关概念来获得短文本的嵌入。
    - 我们进一步将字符级功能集成到模型中，以捕获细粒度的子词信息。
  + 实验结果：在五个常用数据集上的实验结果表明，我们提出的方法明显优于最新方法。
* [【关于 LOTClass】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/classifier_study/LOTClass/)
  + 论文名称：《Text Classification Using Label Names Only: A Language Model Self-Training Approach》
  + 会议：EMNLP2020
  + 论文地址：https://arxiv.org/pdf/2010.07245.pdf
  + 论文源码地址：https://github.com/yumeng5/LOTClass
  + 动机
    - 监督学习：标注数据昂贵
    - 半监督学习：虽然减少了对标注数据的依赖，但还是需要领域专家手动进行标注，特别是在类别数目很大的情况下
    - 关键词积累：关键词在不同上下文中也会代表不同类别
  + 方法：
    - 提出了一种基于预训练神经 LM 的弱监督文本分类模型 LotClass，它不需要任何标记文档，只需要每个类的标签名称。
    - 提出了一种寻找类别指示词的方法和一个基于上下文的单词类别预测任务，该任务训练LM使用一个词的上下文来预测一个词的隐含类别。经过训练的LM很好地推广了基于未标记语料库的自训练文档级分类
  + 在四个分类数据集上，LOTClass明显优于各弱监督模型，并具有与强半监督和监督模型相当的性能。

实战篇

重点推荐篇

* [Transfromer 源码实战](https://github.com/km1994/nlp_paper_study/tree/master/transformer_study/Transformer)
  + [【关于 Transformer 代码实战（文本摘要任务篇）】 那些你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/transformer_study/Transformer/code.md) 【[知乎篇](https://zhuanlan.zhihu.com/p/312044432) 】
* [【关于 Bert 源码解析 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/)
  + [【关于 Bert 源码解析 之 主体篇 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/bertCode1_modeling.md)
  + [【关于 Bert 源码解析 之 预训练篇 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/bertCode2_pretraining.md)
  + [【关于 Bert 源码解析 之 微调篇 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/bertCode3_fineTune.md)
  + [【关于 Bert 源码解析IV 之 句向量生成篇 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/bertCode4_word2embedding.md)
  + [【关于 Bert 源码解析V 之 文本相似度篇 】 那些的你不知道的事](https://github.com/km1994/nlp_paper_study/blob/master/bert_study/T1_bert/bertCode5_similarity.md)

推荐篇

* [爬虫 实战篇](https://github.com/km1994/Conversation_Wp/tree/master/t11_scrapyWp)
  + [Scrapy 爬虫 实战篇](https://github.com/km1994/Conversation_Wp/tree/master/t11_scrapyWp):主要介绍使用 scrapy 构建网络爬虫，并爬去百度搜索引擎数据
* [特征提取 实战篇](https://github.com/km1994/text_feature_extraction)
  + [关键词提取、关键短语提取、文本摘要提取 实战篇](https://github.com/km1994/text_feature_extraction)
  + [TF-idf 特征提取 实战篇](https://github.com/km1994/text_feature_extraction)
  + [pynlp 关键词提取 实战篇](https://github.com/km1994/Conversation_Wp/tree/master/t10_pynlpirWp)
* [词向量预训练 实战篇](https://github.com/km1994/TextClassifier/tree/master/word2vec_train)
  + [word2vec 词向量预训练 实战篇](https://github.com/km1994/TextClassifier/tree/master/word2vec_train)
  + [fasttext 词向量预训练 实战篇](https://github.com/km1994/TextClassifier/tree/master/word2vec_train)
* [中文情感分析 实战篇](https://github.com/km1994/sentiment_analysis)
  + [word2vec](https://github.com/km1994/sentiment_analysis/tree/master/word2vec)
  + [textCNN](https://github.com/km1994/sentiment_analysis/tree/master/textCNN)
  + [charCNN](https://github.com/km1994/sentiment_analysis/tree/master/charCNN)
  + [RCNN](https://github.com/km1994/sentiment_analysis/tree/master/RCNN)
  + [Bi-LSTM](https://github.com/km1994/sentiment_analysis/tree/master/Bi-LSTM)
  + [Bi-LSTM+Attention](https://github.com/km1994/sentiment_analysis/tree/master/Bi-LSTM%252BAttention)
  + [adversarialLSTM](https://github.com/km1994/sentiment_analysis/tree/master/adversarialLSTM)
  + [Transformer](https://github.com/km1994/sentiment_analysis/tree/master/Transformer)
  + [ELMo](https://github.com/km1994/sentiment_analysis/tree/master/ELMo)
  + [BERT](https://github.com/km1994/sentiment_analysis/tree/master/BERT)
* [中文文本分类 实战篇](https://github.com/km1994/TextClassifier)
  + [Tensorflow 篇](https://github.com/km1994/TextClassifier)
    - [FastText](https://github.com/km1994/TextClassifier/tree/master/fastTextStudy.ipynb)
    - [TextCNN](https://github.com/km1994/TextClassifier/tree/master/dl_model)
    - [TextRNN](https://github.com/km1994/TextClassifier/tree/master/dl_model)
    - [TextRCNN](https://github.com/km1994/TextClassifier/tree/master/dl_model)
    - [BiLSTMAttention](https://github.com/km1994/TextClassifier/tree/master/dl_model)
    - [AdversarialLSTM](https://github.com/km1994/TextClassifier/tree/master/dl_model)
    - [Transformer](https://github.com/km1994/TextClassifier/tree/master/dl_model)
  + [pytorch 篇](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [FastText](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [TextCNN](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [TextRNN](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [TextRCNN](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [BiLSTMAttention](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [DPCNN](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [AdversarialLSTM](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
    - [Transformer](https://github.com/km1994/Chinese-Text-Classification-Pytorch)
* [命名实体识别 “史诗级” 入门教程](https://github.com/km1994/NERer)
  + [HMM 做命名实体识别](https://github.com/km1994/named_entity_recognition/models/hmm.py)
  + [CRF 做命名实体识别](https://github.com/km1994/named_entity_recognition/models/crf.py)
  + [BiLSTM-CRF 做命名实体识别](https://github.com/km1994/NERer/tree/master/LSTM_IDCNN)
  + [IDCNN-CRF 做命名实体识别](https://github.com/km1994/NERer/tree/master/LSTM_IDCNN)
  + [BERT-CRF 做命名实体识别](https://github.com/km1994/NERer/tree/master/bert_crf)
  + [ALBERT-CRF 做命名实体识别](https://github.com/km1994/NERer/tree/master/albert_crf)
* 知识图谱 实战篇
  + [KBQA-BERT](https://github.com/km1994/KBQA-BERT)
* [问答系统 实战篇](https://github.com/km1994/Conversation_Wp/tree/master/baidu_qa_zh_process)
  + [基于 百度问答 的问答系统](https://github.com/km1994/Conversation_Wp/tree/master/baidu_qa_zh_process/Baidu_WebQA_model.ipynb)
* [文本匹配 实战篇](https://github.com/km1994/TextMatching)
  + [TextMatching](https://github.com/km1994/TextMatching)
  + [TextMatch](https://github.com/km1994/TextMatch)
  + [Text\_Matching（文本匹配算法）](https://github.com/km1994/Text_Matching)
* 预训练模型 实战篇
  + [bert](https://github.com/km1994/bert)
  + [Chinese-PreTrained-XLNet](https://github.com/km1994/Chinese-PreTrained-XLNet)
* 模型蒸馏 实战篇
  + [基于BERT的蒸馏实验](https://github.com/km1994/bert_distill)

视频学习篇

* [CS224n 视频学习篇](https://github.com/km1994/Datawhale_NLP_CS224n)
  + [Lecture 1: Introduction and Word Vectors](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture1)
  + [Lecture 2: Word Vectors and Word Senses](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture2)
  + [Lecture 3: Word Window Classification, Neural Networks, and Matrix Calculus](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture3)
  + [Lecture 4: Backpropagation](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture4)
  + [Lecture 5: Dependency Parsing](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture5)
  + [Lecture 6: Language Models and RNNs](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture6)
  + [Lecture 7: Vanishing Gradients, Fancy RNNs](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture7)
  + [Lecture 8: Translation, Seq2Seq, Attention](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture8)
  + [Lecture 9: Practical Tips for Projects](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture9)
  + [Lecture 10: Question Answering](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture10)
  + [Lecture 11: Convolutional Networks for NLP](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture11)
  + [Lecture 12: Subword Models](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture12)
  + [Lecture 13: Contextual Word Embeddings](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture13)
  + [Lecture 14: Transformers and Self-Attention](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture14)
  + [Lecture 15: Natural Language Generation](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture15)
  + [Lecture 16: Coreference Resolution](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture16)
  + [Lecture 17: Multitask Learning](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture17)
  + [Lecture 18: Constituency Parsing, TreeRNNs](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture18)
  + [Lecture 19: Bias in AI](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture19)
  + [Lecture 20: Future of NLP + Deep Learning](https://github.com/km1994/Datawhale_NLP_CS224n/tree/master/Lecture/Lecture20)

Elastrsearch 学习篇

* [Elastrsearch 学习](es_study/)
  + [ElasticSearch架构解析与最佳实践.md](es_study/ElasticSearch%E6%9E%B6%E6%9E%84%E8%A7%A3%E6%9E%90%E4%B8%8E%E6%9C%80%E4%BD%B3%E5%AE%9E%E8%B7%B5.md)

推荐系统 学习篇

* [推荐系统 基础](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study)
  + [【关于 推荐系统】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/base1_%E5%9F%BA%E7%A1%80%E6%A6%82%E5%BF%B5%E7%AF%87.md)
  + [【关于 召回】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/base2_%E5%8F%AC%E5%9B%9E%E7%AF%87.md)
  + [【关于 embedding召回】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/base3_embedding%E5%8F%AC%E5%9B%9E.md)
  + [【关于 协同过滤】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/)
  + [【关于 矩阵分解】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/base5_%E7%9F%A9%E9%98%B5%E5%88%86%E8%A7%A3.md)
  + [【关于 FM】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/base_study/base6_FM.md)
* [推荐系统 论文学习](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study)
  + [DeepFM 论文学习](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/DeepFM_study)
  + [DeepWalk 论文学习](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/DeepWalk)
  + [ESMM 论文学习](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/ESMM_study)
  + [【关于 FiBiNET】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/FiBiNet_study)
  + [【关于 DeepCF】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/RecommendedSystem_study/DeepCF_study)

竞赛篇

* [竞赛篇](game_study)

[GCN\_study学习篇](https://github.com/km1994/GCN_study)

* GCN 介绍篇
  + [Graph 介绍](https://github.com/km1994/GCN_study/blob/master/graph_introduction/graph_introduction.md)
  + [Weisfeiler-Leman 算法介绍](https://github.com/km1994/GCN_study/blob/master/WL/WL_conclusion.md)
* GCN 三剑客
  + [Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering](https://github.com/km1994/GCN_study/blob/master/CNNonGraph_Defferrard_2016/CNNonGraph_Defferrard_2016_%E6%80%BB%E7%BB%93.md)
  + [SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS](https://github.com/km1994/GCN_study/blob/master/GCN/SEMI-SUPERVISED%2520CLASSIFICATION%2520WITH%2520GRAPH%2520CONVOLUTIONAL%2520NETWORKS.pdf)
  + [Attention Models in Graphs: A Survey](https://github.com/km1994/GCN_study/blob/master/Attention_models/Attention_models.md)
* 经典篇
  + [Can GNN go “online”?an analysis of pretraining and inference](https://github.com/km1994/GCN_study/tree/master/CanGNNGoOnline)
  + [Graph Convolutional Networks for Text Classification](https://github.com/km1994/GCN_study/tree/master/GCNforTextClassification)
  + [HOW POWERFUL ARE GRAPH NEURAL NETWORKS](https://github.com/km1994/GCN_study/blob/master/HowPowerGCN/HOW%2520POWERFUL%2520ARE%2520GRAPH%2520NEURAL%2520NETWORKS.pdf)
  + [Graph Convolutional Matrix Completion](https://github.com/km1994/GCN_study/blob/master/GCMC/Graph%2520Convolutional%2520Matrix%2520Completion.pdf)
  + [Representation Learning For Attributed Multiplex Heterogeneous Network](https://github.com/km1994/GCN_study/blob/master/RepresentationLearningForAttributedMultiplexHeterogeneousNetwork/RepresentationLearningForAttributedMultiplexHeterogeneousNetwork.md)
* 预训练篇
  + [GNN 教程：GCN 的无监督预训练](https://github.com/km1994/GCN_study/tree/master/pretrainingGCN_1)
  + [Pre-training Graph Neural Networks](https://github.com/km1994/GCN_study/blob/master/pretrainingGCN_2/pretrainingGCN.md)
* 实战篇
  + [DGL](https://github.com/km1994/dgl)
  + [DGL 入门](https://github.com/km1994/GCN_study/blob/master/DGL_study/DGL_introduction.md)
  + [DGL 入门 —— GCN 实现](https://github.com/km1994/GCN_study/blob/master/DGL_study/DGL_GCN_introduction.md)

[ML 小白入门篇](https://github.com/km1994/ML_beginer)

* [【关于 Logistic Regression 】那些你不知道的事](https://github.com/km1994/nlp_paper_study/tree/master/ML_study/LogisticRegression_study/)
* [概率图模型串烧 （HMM->MEMM->CRF）](https://github.com/km1994/ML_beginer/tree/master/CRF_study)
* [KNN 算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25B8%2580knnstudy)
  + [理论篇](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100000006&idx=7&sn=f040e0a95376880349378ce7afd634af&chksm=1bbfe43c2cc86d2a5942e0938c871375cffcd72c9795a3e8b39fdb0c8285348631e7bf8a27b4)
  + [实战篇](https://github.com/km1994/MLStudy/blob/master/KNNStudy/KNNClass.py)
* [朴素贝叶斯算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25BA%258Cnbstudy)
  + [NB 算法理论学习](https://github.com/km1994/MLStudy/blob/master)
  + [NB 算法实现](https://github.com/km1994/MLStudy/blob/master/NBStudy/NBStudy.py)
* [Apriori 算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25B8%2589aprioristudy)
  + [Apriori 算法理论学习](https://github.com/km1994/MLStudy/blob/master)
  + [Apriori 算法实现](https://github.com/km1994/MLStudy/blob/master/AprioriStudy/AprioriMyTest.py)
* [Softmax 算法学习篇](https://github.com/km1994/MLStudy#%25E5%258D%2581softmax-numpy-%25E5%25AE%259E%25E7%258E%25B0)
  + [Softmax 理论学习](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100001925&idx=5&sn=20c5ead4f4b5f8f88c30043fb3703557&chksm=1bbfedbf2cc864a96b5fc4575e09294478f6f4cff65d654b8d775fed78766f80faf333d8ca08&scene=20&xtrack=1#rd)
  + [Softmax 算法实现](https://github.com/km1994/MLStudy/blob/master/softmaxStudy/softmaxStudy.py)
* [Gradient Descent 算法学习篇](https://github.com/km1994/MLStudy#%25E5%259B%259Bgradientdescentstudy)
  + [GradientDescent 算法理论学习](https://github.com/km1994/MLStudy/blob/master)
  + [GradientDescent 算法实现](https://github.com/km1994/MLStudy/blob/master/GradientDescentStudy/GradientDescentTest.py)
* [随机森林算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25BA%2594randomforest)
  + [RandomForest 算法理论学习](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100001388&idx=1&sn=21bae727bf3510fad98b3ec4a89d124e&chksm=1bbfe3562cc86a40769ea726f96e3a45185697f9582a2e3fbbbeec3af90dd722ebe09b635ddc&scene=20&xtrack=1#rd)
  + [RandomForest 算法实现](https://github.com/km1994/MLStudy/blob/master/RandomForest/xiechengRF/RandomForestClass.py)
  + [基于PCA 的 RandomForest 算法实现](https://github.com/km1994/MLStudy/blob/master/RandomForest/xiechengRF/xiechengPCARF.py)
* [EM 算法学习篇](https://github.com/km1994/ML_beginer/tree/master/EM_study)
* [SVM 算法学习篇](https://github.com/km1994/MLStudy#%25E5%2585%25ADsvn)
  + [SVN 算法理论学习](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100001388&idx=1&sn=21bae727bf3510fad98b3ec4a89d124e&chksm=1bbfe3562cc86a40769ea726f96e3a45185697f9582a2e3fbbbeec3af90dd722ebe09b635ddc&scene=20&xtrack=1#rd)
  + [SVM 算法学习篇](https://github.com/km1994/ML_beginer/tree/master/SVM_study)
  + [SVN 算法实现](https://github.com/km1994/MLStudy/blob/master/SVN/PCAandSVN.py)
* [BPNN 算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25B8%2583bpnn)
  + [BPNN 算法理论学习](https://github.com/km1994/MLStudy/blob/master)
  + [BPNN 算法实现](https://github.com/km1994/MLStudy/blob/master/ANN/BP/test.py)
* [PCA 算法 学习篇](https://github.com/km1994/MLStudy#%25E5%2585%25ABpca)
  + [PCA 算法理论学习](https://github.com/km1994/MLStudy/blob/master)
  + [PCA 算法实现](https://github.com/km1994/MLStudy/blob/master/DimensionalityReduction/PCA/PCAClass.py)
* [CNN 算法 学习篇](https://github.com/km1994/MLStudy#%25E4%25B9%259Dcnn-numpy-%25E5%25AE%259E%25E7%258E%25B0)
  + [卷积运算的定义、动机](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100000973&idx=2&sn=4eced9e7204274a4e3798cb73a140c72&chksm=1bbfe1f72cc868e1f63262fad39b4f735a6d424c064f6bee755e438b94487bf75b5d41cc02c0&scene=20&xtrack=1&key=fe048f5ad4fa1bcff1ed72e320faab18cb01c02c1a16279c60775734b428e42206e9f5a8f3c402ae96c01259df639ca04206da43e2ab1b42bfaf44bb4068c793df27faa893ea0301a375086e4adfd3b7&ascene=1&uin=MjQ3NDIwNTMxNw==&devicetype=Windows+10&version=62060426&lang=zh_CN&pass_ticket=906xy/k9TQJp5jnyekYc89wLa17ODmZRkas9HXdX+tYcy0q32NIxLHOhFx519Yxa)
  + [反卷积Deconvolution](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100000973&idx=3&sn=8a787cc0e165fa071ca7a602f16fae17&chksm=1bbfe1f72cc868e1249a3ebe90021e2a6e12c3d8021fcc1877a5390eed36f5a8a6698eb65216&scene=20&xtrack=1#rd)
  + [池化运算的定义](https://mp.weixin.qq.com/s?__biz=MzAxMTU5Njg4NQ==&mid=100000973&idx=4&sn=cebf71790dd7e0e497fa36fa199c368d&chksm=1bbfe1f72cc868e1017d26a996f1eb7602fad1efced62713def3012b8df1b85b6ba46c0ebae8&scene=20&xtrack=1#rd)
  + [CNN 算法 numpy 实现](https://github.com/km1994/MLStudy/blob/master/CNN-Numpy_suss/view.py)

[资源篇](https://github.com/km1994/funNLP)

* [funNLP](https://github.com/km1994/funNLP)

[CV 入门 实战篇](https://github.com/km1994/cv_entrance)