共性 AI 算法平台构建-NLP

严禁复制

1 文本表示

1.1 静态词向量

1.1.1 word2vec

下载地址：https://github.com/danielfrg/word2vec

算法描述：

Word2Vec 是 google 在 2013 年推出的一个 NLP 工具，它的特点是能够将单词转化为向量来表示，这样词与词之间就可以定量的去度量他们之间的关系，挖掘词之间的联系。Word2Vec 包括 CBOW 和 Skip-gram 两种模型，CBOW 适合于数据集较小的情况，而 Skip-Gram 在大型语料中表现更好。其与 CBOW 根据语境预测目标单词不同，Skip-gram 根据当前单词预测语境。

1.1.2 GloVe

下载地址：https://github.com/stanfordnlp/GloVe

算法描述：

GloVe（Global Vectors for Word Representation）是一种用于学习单词向量的无监督学习算法。它通过在全局上下文中对单词的共现信息进行建模，生成了词向量表示。GloVe 旨在捕捉单词之间的语义关系，类似于 Word2Vec。

GloVe 的训练目标是学习词向量，使其点积等于词共现概率的对数。由于比率的对数等于对数的差值，因此该目标将共现概率的比率（对数）与词向量空间中的向量差相关联。由于这些比率可以编码某种形式的含义，因此此信息也被编码为向量差异。出于这个原因，生成的词向量在词类比任务上表现非常好，例如在 word2vec 包中检查的任务。

1.1.3 FastText

下载地址：https://github.com/facebookresearch/fastText

算法描述：

FastText 是一种用于学习文本表示的快速文本分类器和词向量学习算法。它是由 Facebook AI Research (FAIR)实验室开发的，并以其速度和在处理子词（subword）信息方面的能力而闻名。FastText 的主要目标之一是在大规模文本数据上进行快速而有效的训练，同时考虑到了子词信息。与 Word2Vec 不同，FastText 通过将单词表示为其字符 n-gram 的平均值来处理单词的子词信息。FastText 可以学习到单词的分布式表示，即使单词不存在于训练数据中，也能够通过其子词信息进行推断。

1.2 动态词向量

1.2.1 ELMo 方法

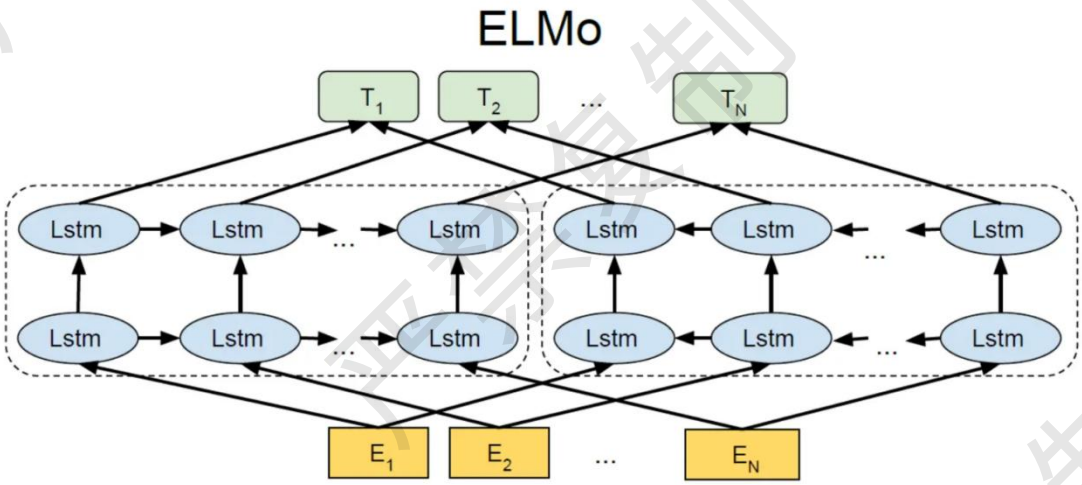
项目地址：https://github.com/allenai/allennlp-models

https://allenai.org/allennlp/software/elmo

算法描述：

ELMo（Embeddings from Language Models）是一种动态词向量模型，由 Allen Institute for Artificial Intelligence (AI2)开发。相比于传统的静态词向量，ELMo 考虑了上下文信息，并生成了基于上下文的动态词向量表示，这使得 ELMo 能够更好地捕捉单词的语义变化和多义性。

ELMo 首先在大型的数据集上训练好模型，然后再后续任务中可以根据输入的句子，输出每一个单词的词向量。ELMo 在结构上使用了双向的 LSTM（Long Short-Term Memory）神经网络结构，通过从左到右和从右到左两个方向来学习上下文信息。ELMo 模型是一个深层的双向 LSTM 模型，其中每一层都会输出一个词的表示。



评估结果：

将 ELMo 添加到现有的 NLP 系统中，显著提高了每一项考虑到的任务的最先进水平。在大多数情况下，它们可以简单地交换为预先训练的 GloVe 或其他单词向量。

Task	Previous SOTA	Our baseline	ELMo + Baseline	Increase (Absolute/Relative)
SQuAD	SAN84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al (2017)88.6	88.0	88.7 +/- 0.17	0.7 / 5.8%
SRL	He et al (2017)81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al (2017)67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al (2017)91.93 +/- 0.19	90.15	92.22 +/- 0.10	2.06 / 21%

Sentiment (5-class)	McCann et al (2017)53.7	51.4	54.7 +/- 0.5	3.3 / 6.8%

1.2.2 Bert 类方法

项目地址：https://github.com/google-research/bert

模型描述：

BERT（Bidirectional Encoder Representations from Transformers）是一种革命性的自然语言处理（NLP）模型，于 2018 年由 Google 提出。BERT 的关键创新在于使用了 Transformer 模型，并引入了双向（bidirectional）的预训练策略，从而更全面地理解上下文信息。

BERT 的预训练过程分为两个阶段：首先是无监督的预训练，模型在大规模文本数据上学习语言模型，其次是有监督的微调，模型在特定任务上进行监督学习。在无监督预训练中，BERT 通过遮蔽语言模型（Masked Language Model, MLM）的方式，随机遮蔽输入文本中的一些单词，然后预测这些被遮蔽的单词。这使得 BERT 在学习上下文信息时能够考虑到句子中的所有单词，而不仅仅是上文或下文。

BERT 之所以强大，是因为它生成了丰富的上下文相关的词嵌入表示。在进行下游任务时，可以将 BERT 的预训练模型作为特征提取器，或者通过微调在特定任务上进一步优化模型参数。BERT 在多个 NLP 任务上取得了领先的性能，如文本分类、命名实体识别、问答等。

1.2.3 GPT 类方法

下载地址：https://huggingface.co/openai-community/gpt2

模型描述：

GPT（Generative Pre-trained Transformer）是一系列基于 Transformer 架构的预训练模型，由 OpenAI 提出。GPT 的关键创新在于采用了自回归（autoregressive）的预训练策略，即模型通过生成下一个词来预测文本序列。该系列包括了多个版本，其中 GPT-3 是目前规模最大的版本。

GPT 的预训练过程是在大规模文本数据上进行的，模型学会了语言的统计结构和丰富的上下文信息。在预训练完成后，GPT 可以用于各种下游任务，如文本生成、文本分类、问答等。在应用阶段，通过微调或者直接使用 GPT 生成的文本，可以实现强大的自然语言处理能力。

GPT 的模型结构包含了多个层的 Transformer 编码器，每一层都具有多头自注意力机制。在预训练时，GPT 通过最大似然估计的方式，优化模型参数，使得模型在生成文本时能够

更好地捕捉上下文信息。GPT 系列模型的强大之处在于其能够生成连贯、富有语义的文本，且不需要任务特定的标签信息。

1.3 中文通用文本表示模型

1.3.1 CoROM 文本向量-中文-通用领域-base

项目地址：

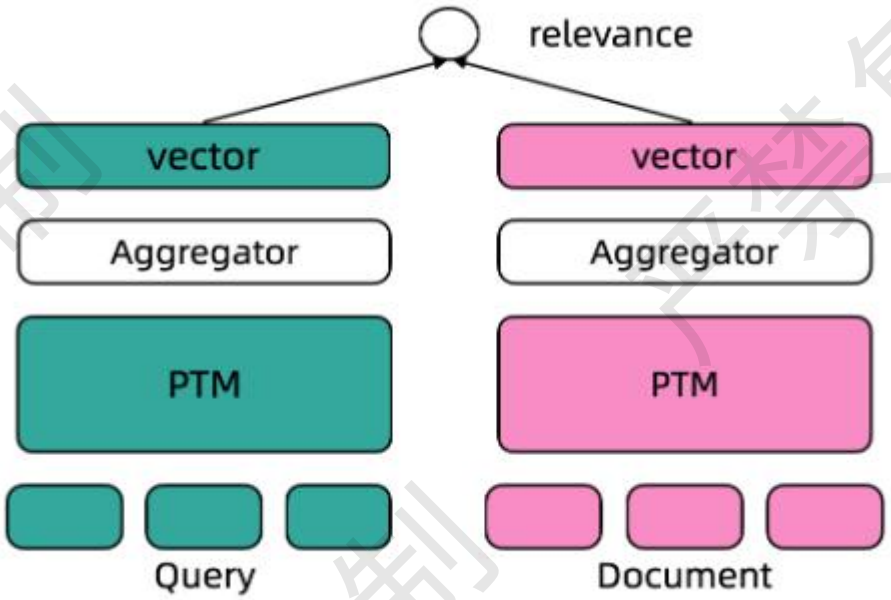
https://www.modelscope.cn/models/damo/nlp_corom_sentence-embedding_chinese-base/summary

模型描述：

CoRom 模型是一种基于双塔结构的检索模型，它将 query 和 passage 分别输入到两个独立的神经网络中进行编码，得到对应的向量表示，然后通过计算向量之间的相似度来进行检索。在 CoRom 模型中，query 和 passage 的向量表示是分别计算的，因此它们在代码中的处理方式也是不同的。具体来说：对于 query，模型会将其传入 model.forward 函数的 source_sentence 参数中，然后使用 model.encode_query 函数对其进行编码，得到对应的向量表示。对于 passage，模型会将其传入 model.forward 函数的 sentences_to_compare 参数中，然后使用 model.encode_sentences 函数对其进行编码，得到对应的向量表示。

Dual Encoder 文本表示模型：

基于监督数据训练的文本表示模型通常采用 Dual Encoder 框架，如下图所示。在 Dual Encoder 框架中，Query 和 Document 文本通过预训练语言模型编码后，通常采用预训练语言模型[CLS]位置的向量作为最终的文本向量表示。基于标注数据的标签，通过计算 query-document 之间的 cosine 距离度量两者之间的相关性。



CoRom 模型可以使用在通用领域的文本向量表示及其下游应用场景，包括双句文本相似度计算、query&多 doc 候选的相似度排序。

评估结果：

在文本向量召回场景下评估模型效果, DuReader Retrieval 召回评估结果如下:

Model	MRR@10	Recall@1	Recall@50
BM25	21.97	12.85	66.35
DPR	60.45	45.75	91.75
CoROM-Base	65.82	54.68	93.00
CoROM-Tiny	34.90	24.65	77.63

1.3.2 Udever 多语言通用文本表示模型 3b

下载地址：https://modelscope.cn/models/damo/udever-bloom-3b/summary

模型描述：

Udever 模型是基于 BLOOM 系列模型在 MS MARCO passage + SNLI + MNLI 三个数据上 bitfit 微调得到的文本表示模型，可用于文本检索、代码检索、文本相似性等多种任务。仅使用英文文本数据进行对比学习微调的 Udever 在英文、中文等多种自然语言和 Python、Java 等多种编程语言都有很好的表现。可视化的例子见图 1。

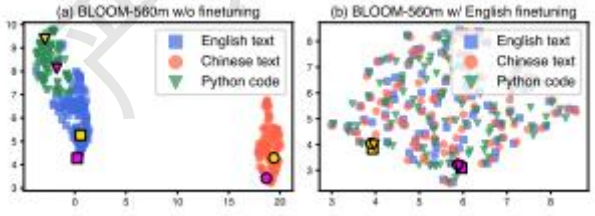


Figure 1: Visualization of 100 examples from CodeSearchNet Python, where Chinese texts are translated by GPT-3.5-turbo. Gold and pink markers represent parallel sequences in different languages. Before finetuning, (a), embeddings are separated by language, especially English and Chinese. After English finetuning, (b), the parallel sequences are well aligned to each other.

上图 1 中是英文、中文、python 代码的 embedding 可视化，其中蓝色正方形代表英文文本注释，绿色三角形代表 python 代码，红色圆形代表翻译的中文注释文本，粉色和黄色的图标三元组代表两组语义一致的三元组，即[英文注释，对应的 python 代码，翻译的中文注释]。

- 左图是原始 bloom 模型使用 SGPT 方式产生的向量（embedding），不同语言的向量分

布在不同的区域；黄色和粉色的三元组是分散的。

- 右图是 Udever-bloom-560m 模型的向量，不同语言的向量按照语义均匀分布；黄色的三元组按照语义分布在了一起，粉色也是。

本模型基于 MS MARCO passage + SNLI + MNLI 三个数据集(通用领域)上训练，在垂直领域文本上的效果为无监督水平。

评估结果：

表 1: 英文多任务向量评测榜单 MTEB 结果

MTEB	Avg.	Class.	Clust.	PairClass.	Rerank.	Retr.	STS	Summ.
#Datasets (→\rightarrow→)	56	12	11	3	4	15	10	1
bge-large-en-v1.5	64.23	75.97	46.08	87.12	60.03	54.29	83.11	31.61
bge-base-en-v1.5	63.55	75.53	45.77	86.55	58.86	53.25	82.4	31.07
gte-large	63.13	73.33	46.84	85	59.13	52.22	83.35	31.66
gte-base	62.39	73.01	46.2	84.57	58.61	51.14	82.3	31.17
e5-large-v2	62.25	75.24	44.49	86.03	56.61	50.56	82.05	30.19
instructor-xl	61.79	73.12	44.74	86.62	57.29	49.26	83.06	32.32
instructor-large	61.59	73.86	45.29	85.89	57.54	47.57	83.15	31.84
e5-base-v2	61.5	73.84	43.8	85.73	55.91	50.29	81.05	30.28
e5-large	61.42	73.14	43.33	85.94	56.53	49.99	82.06	30.97
text-embedding-ada-002 (OpenAI API)	60.99	70.93	45.9	84.89	56.32	49.25	80.97	30.8
e5-base	60.44	72.63	42.11	85.09	55.7	48.75	80.96	31.01
SGPT-5.8B-msmarco	58.93	68.13	40.34	82	56.56	50.25	78.1	31.46
sgpt-bloom-7b1-msmarco	57.59	66.19	38.93	81.9	55.65	48.22	77.74	33.6
Udever-bloom-560m	55.80	68.04	36.89	81.05	52.60	41.19	79.93	32.06
Udever-bloom-1b1	58.28	70.18	39.11	83.11	54.28	45.27	81.52	31.10
Udever-bloom-3b	59.86	71.91	40.74	84.06	54.90	47.67	82.37	30.62

MTEB	Avg.	Class.	Clust.	PairClass.	Rerank.	Retr.	STS	Summ.
Udever-bloom-7b1	60.63	72.13	40.81	85.40	55.91	49.34	83.01	30.97

表 2: CodeSearchNet 文本搜索代码评测

CodeSearchNet	Go	Ruby	Python	Java	JS	PHP	Avg.
CodeBERT	69.3	70.6	84.0	86.8	74.8	70.6	76.0
GraphCodeBERT	84.1	73.2	87.9	75.7	71.1	72.5	77.4
cpt-code S	97.7	86.3	99.8	94.0	86.0	96.7	93.4
cpt-code M	97.5	85.5	99.9	94.4	86.5	97.2	93.5
sgpt-bloom-7b1-msmarco	76.79	69.25	95.68	77.93	70.35	73.45	77.24
Udever-bloom-560m	75.38	66.67	96.23	78.99	69.39	73.69	76.73
Udever-bloom-1b1	78.76	72.85	97.67	82.77	74.38	78.97	80.90
Udever-bloom-3b	80.63	75.40	98.02	83.88	76.18	79.67	82.29
Udever-bloom-7b1	79.37	76.59	98.38	84.68	77.49	80.03	82.76

表 3: Multi-cpr 中文特殊领域检索评测结果

			E-commerce		Entertainment video		Medical	
Model	Train	Backbone	MRR@10	Recall@1k	MRR@10	Recall@1k	MRR@10	Recall@1k
BM25	-	-	0.225	0.815	0.225	0.780	0.187	0.482
Doc2Query	-	-	0.239	0.826	0.238	0.794	0.210	0.505
DPR-1	In-Domain	BERT	0.270	0.921	0.254	0.934	0.327	0.747
DPR-2	In-Domain	BERT-CT	0.289	0.926	0.263	0.935	0.339	0.769
text-embedding-ada-002	General	GPT	0.183	0.825	0.159	0.786	0.245	0.593
sgpt-bloom-7b1-msm	General	BLOO	0.242	0.840	0.227	0.829	0.311	0.675

严禁复制

			E-commerce		Entertainment video		Medical	
arco		M						
Udever-bloom-560m	General	BLOOM	0.156	0.802	0.149	0.749	0.245	0.571
Udever-bloom-1b1	General	BLOOM	0.244	0.863	0.208	0.815	0.241	0.557
Udever-bloom-3b	General	BLOOM	0.267	0.871	0.228	0.836	0.288	0.619
Udever-bloom-7b1	General	BLOOM	0.296	0.889	0.267	0.907	0.343	0.705

2 文本纠错

2.1 文本纠错的语料库

2.1.1 中文拼写纠错

（1）SIGHAN 系列

来自于 SIGHAN2013、SIGHAN2014、SIGHAN2015 评测任务，分别包括 350/6526/3174 句训练集和 974/526/550 句测试集。

下载地址：https://github.com/onebula/sighan_raw

（2）OCR dataset

爱奇艺基于 OCR 技术生成的伪 CSC 训练集，4575 句。

下载地址：https://github.com/iqiyi/FASPell

（3）Hybrid Dataset

基于 OCR 和 ASR 技术伪造的 CSC 训练集，约 27 万条。

下载地址：https://github.com/wdimmy/Automatic-Corpus-Generation

（4）ESpell

苏州大学开放的多领域拼写纠错数据集，包括金融、医药等领域。

下载地址：https://github.com/Aopolin-Lv/ECSpell

2.1.2 中文语法纠错

（1）CGED

北京语言大学团队开放的 CGED 系列数据集，面向二语者文本，早期仅包含语法错误检测任务，后期包含语法错误纠正任务。

下载地址：https://github.com/wdimmy/Automatic-Corpus-Generation

（2）NLPCC18

北京大学团队于 NLPCC2018 上开放的语法纠错评测任务数据集，面向二语者文本，同期开放的还有 Lang8 训练数据集。

下载地址：http://tcci.ccf.org.cn/conference/2018/dldoc/trainingdata02.tar.gz

（3）MuCGEC

苏州大学和阿里巴巴达摩院开放的 CGEC 数据集，面向二语者文本，包含 3 个领域和多答案。

下载地址：https://github.com/HillZhang1999/MuCGEC

2.2 单词拼写纠错

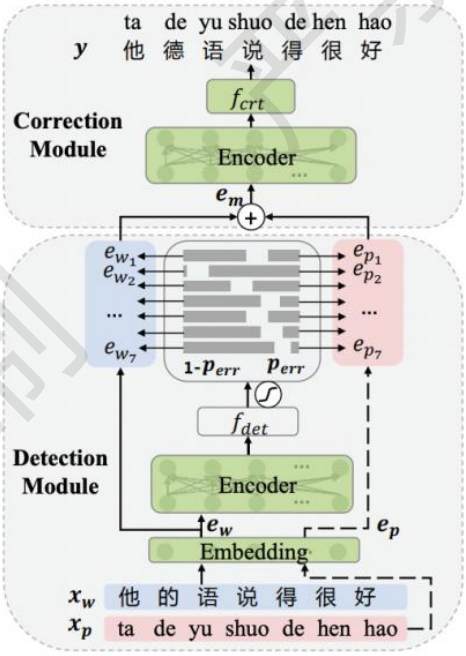
单词拼写纠错是一种常见的自然语言处理任务。拼写纠错算法的实现通常包括两个子任务：拼写错误检测和拼写错误纠正。拼写错误检测可以分为两种类型：非单词错误和真实单词错误。非单词错误指的是那些拼写错误后的词本身就不合法，如错误的将“giraffe”写成“graffe”；真实单词错误指的是那些拼写错误后的词仍然是合法的情况，如将“there”错误拼写为“three”（形近），将“peace”错误拼写为“piece”（同音），将“two”错误拼写为“too”（同音）。拼写错误纠正的目标是自动纠正拼写错误，如把“hte”自动校正为“the”，或者给出一个最可能的拼写建议，甚至一个拼写建议列表。

2.2.1 ernie-csc

项目地址：https://github.com/orangetwo/ernie-csc

算法描述：ERNIE-CSC 是一种用于中文文本纠错的算法。它是百度在 ACL 2021 上提出的一种基于拼音特征的 Softmask 策略的中文错别字纠错的下游任务网络。 ERNIE-CSC 的目标是自动检测和纠正中文文本中的语法错误，包括多字、少字、错别字等。ERNIE-CSC 使用了噪声信道模型，这是一种普适性的模型，被用于语音识别、拼写纠错、机器翻译、中文分词、词性标注、音字转换等众多应用领域， ERNIE-CSC 的实现方法包括字典匹配、编辑距离、噪声信道模型等。

端到端文本纠错包括 Detection Module 和 Correction Module 两个部分。模型图示意如下图所示：



该模型在 SIGHAN 简体版数据集以及 Automatic Corpus Generation 生成的中文纠错数据集生成的中文纠错数据集上进行 Finetune 训练。本仓库已经把原始的语料进行处理，即可以直接用本仓库提供的语料进行训练。

2.2.2 FASPell

项目地址：https://github.com/iqiyi/FASPell

算法描述：FASPell 提出了一种解决中文拼写错误的新范式，抛弃了传统的混淆集转而训练了一个以 BERT 为基础的深度降噪编码器(DAE)和以置信度-字音字形相似度为基础的解码器(CSD)。通过使用 BERT 可以动态的生成候选集，通过 CSD 解码器（置信度-字音字形相似度）可以有效提高纠错效果。

模型主要分两大块组成：

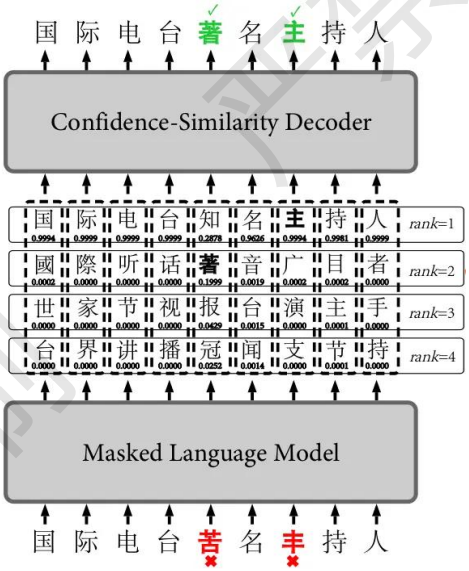
第一：Masked Language Model (bert)

是一个自动编码器（DAE), 基于 bert 模型，每次获取预测词的 top k 个候选字。

第二：Cofidence-Similarity Decoder

该部分是一个解码器，通过编码器输出的置信度 confidence 分值和中文字音字形的相似度 similarity 分值两个维度进行候选集的过滤和刷选，选择最佳候选的路径作为输出。

其整体架构如下图所示：



FASPell 在 SIGHAN15 测试集上的性能表现如下：

sentence-level 性能：

	Precision	Recall
Detection	67.6%	60.0%
Correction	66.6%	59.1%

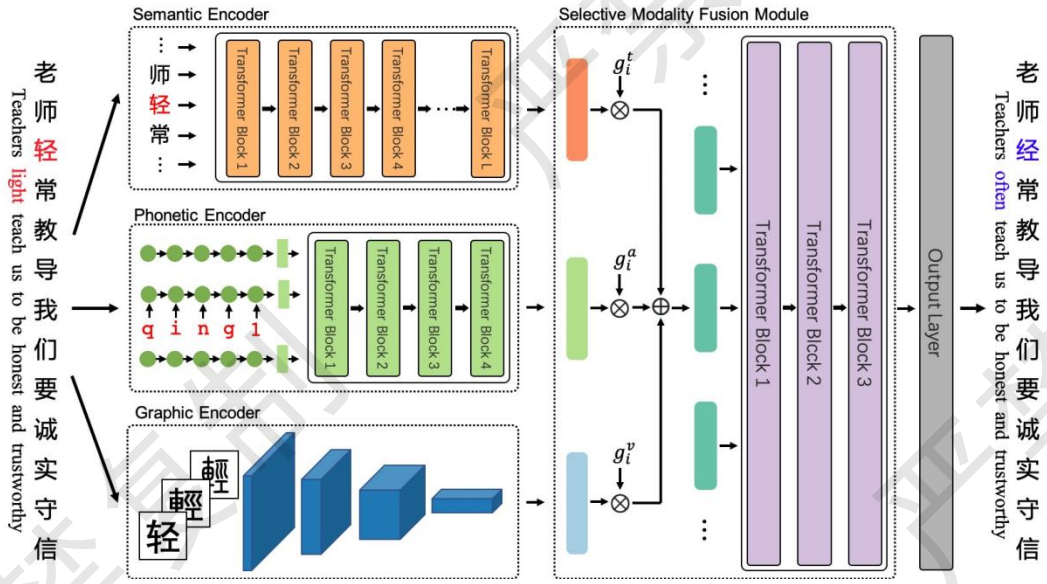
character-level 性能:

	Precision	Recall
Detection	76.2%	67.1%
Correction	73.5%	64.8%

2.2.3 Realise

项目地址：https://github.com/DaDaMrX/ReaLiSe

算法描述：Realise 模型是一个多模态的中文拼写检查模型。该模型使用特定的语义、语音和图形编码器捕捉这些形式的信息，并提出一种选择性模态融合机制控制这些模态的信息流。SIGHAN 基准显示，提出的算法比仅适用文本信息的基线模型具有更大优势，使用听觉和视觉信息有助于提高拼写检查的准确性。

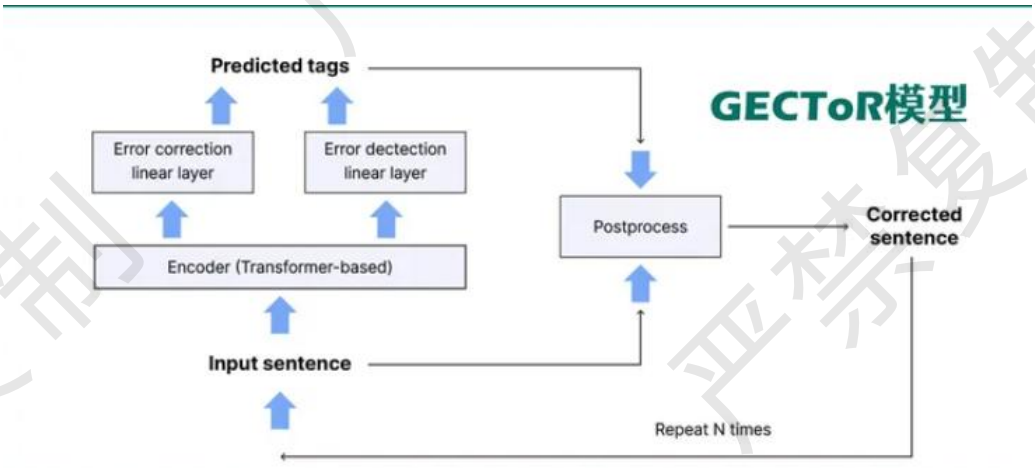


2.3 语法使用纠错

2.3.1 Seq2Edit

项目地址：https://github.com/grammarly/gector

算法描述：基于 Seq2Edit 的语法纠错模型已经被广泛应用在中英文语法纠错任务中，特点是速度快（非自回归），修改精度高。例如 Grammarly 开源的[2]。GECToR 模型本质上是一个序列标注模型，它的解码空间是插入、删除、替换等编辑操作。通过并行预测编辑并将其应用于原句子，GECToR 模型能够完成长度可变的语法纠错。上述过程也可以多轮迭代进行，从而进一步提升修改效果。

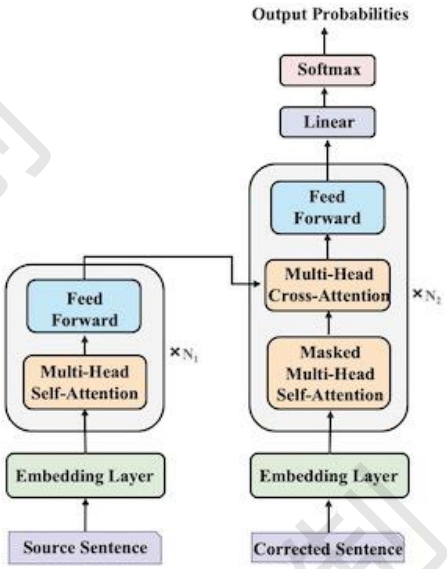


2.3.2 Seq2Seq

项目地址：

https://modelscope.cn/models/damo/nlp_bart_text-error-correction_chinese/summary

算法描述：输入一句中文文本，文本纠错技术对句子中存在拼写、语法、语义等错误进行自动纠正，输出纠正后的文本。采用基于 transformer 的 seq2seq 方法建模文本纠错任务。模型训练上，我们使用中文 BART 作为预训练模型，然后在 Lang8 和 HSK 训练数据上进行 finetune。



2.4 中文文本纠错

中文文本纠错论文汇总：

https://github.com/nghuyong/Chinese-text-correction-papers

2.4.1 BART 文本纠错-中文-通用领域-large

项目地址：

https://modelscope.cn/models/damo/nlp_bart_text-error-correction_chinese/summary

算法描述： 输入一句中文文本，文本纠错技术对句子中存在拼写、语法、语义等错误进行自动纠正，输出纠正后的文本。主流的方法为 seq2seq 和 seq2edits，采用基于 transformer 的 seq2seq 方法建模文本纠错任务。模型训练上，我们使用中文 BART 作为预训练模型，然后在 Lang8 和 HSK 训练数据上进行 finetune。不引入额外资源的情况下，

本模型在 NLPCC18 测试集上，采用 M2ScorerNLPCC18 官方评测工具评估，同等规模和训练数据的模型中取得了 SOTA。

	P	R	F0.5
Tang et al., 20211	47.41	23.72	39.51
Sun et al., 20212	45.33	27.61	40.17
Ours3	48.89	32.80	44.53

2.4.2 SpellGCN

项目地址：https://github.com/ACL2020SpellGCN/SpellGCN

算法描述：主要通过 graph convolutional network (GCN）对字音和字形结构关系进行学习，并且将这种字音字形的向量融入到字的 embedding 中，在纠错分类的时候，纠错更倾向于预测为混淆集里的字。模型训练是一个 end-to-end 的过程，试验显示，在公开的中文纠错数据集上有一个较大的提升。

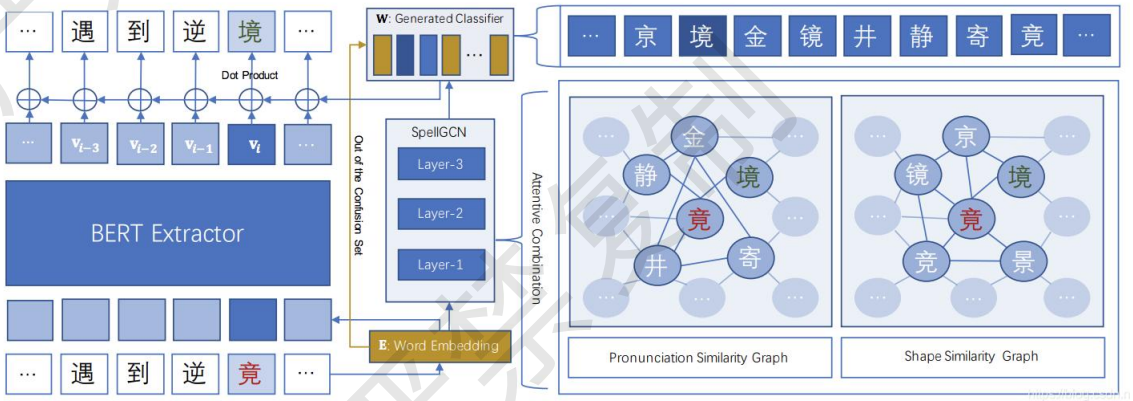
模型也主要分两部分组成：

第一部分：特征提取器

特征提取器基于 12 层的 bert 最后一层的输出

第二部分：纠错分类模型

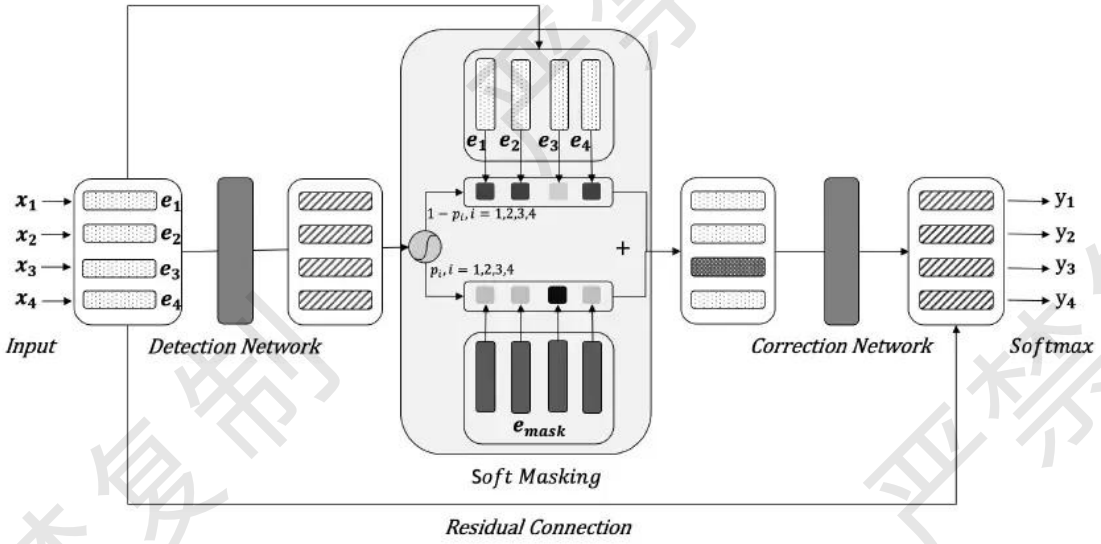
通过 GCN 学习字音字形相似结构信息，融合字的语义信息和字的结构信息，在分类层提高纠错准确率。



2.4.3 Soft-Mask BERT

项目地址：https://github.com/wanglke/Soft-Masked-BERT

算法描述：Soft-Mask BERT 将纠错任务分成两部分：detection network（错误检测）和 correction network（错误纠正）。在错误检测部分，通过 BiGRU 模型对每个输入字符进行错误检测，得到每个输入字符的错误概率值参与计算 soft-masked embedding 作为纠错部分的输入向量，一定程度减少了 bert 模型的过纠问题，提高纠错准确率。



模型 Input 是字粒度的 word-embedding，可以使用 BERT-Embedding 层的输出或者 word2vec。检测网络由 Bi-GRU 组成，充分学习输入的上下文信息，输出是每个位置 i 可能为错别字的概率 p(i)，值越大表示该位置出错的可能性越大。

2.4.4 Pycorrector

pycorrector 是一个中文文本纠错工具。支持中文音似、形似、语法错误纠正，python3.8 开发。。pycorrector 实现了 Kenlm、ConvSeq2Seq、BERT、MacBERT、ELECTRA、ERNIE、Transformer 等多种模型的文本纠错，并在 SigHAN 数据集评估各模型的效果。

模型：

- Kenlm 模型：本项目基于 Kenlm 统计语言模型工具训练了中文 NGram 语言模型，结合规则方法、混淆集可以纠正中文拼写错误，方法速度快，扩展性强，效果一般。
- DeepContext 模型：本项目基于 PyTorch 实现了用于文本纠错的 DeepContext 模型，该模型结构参考 Stanford University 的 NLC 模型，2014 英文纠错比赛得第一名，效果一般。
- Seq2Seq 模型：本项目基于 PyTorch 实现了用于中文文本纠错的 ConvSeq2Seq 模型，该模型在 NLPCC-2018 的中文语法纠错比赛中，使用单模型并取得第三名，可以并行训练，模型收敛快，效果一般。
- T5 模型：本项目基于 PyTorch 实现了用于中文文本纠错的 T5 模型，使用 Langboat/mengzi-t5-base 的预训练模型 finetune 中文纠错数据集，模型改造的潜力较大，效果好。
- ERNIE_CSC 模型：本项目基于 PaddlePaddle 实现了用于中文文本纠错的 ERNIE_CSC 模型，模型在 ERNIE-1.0 上 finetune，模型结构适配了中文拼写纠错任务，效果好。

- MacBERT 模型【推荐】: 本项目基于 PyTorch 实现了用于中文文本纠错的 MacBERT4CSC 模型，模型加入了错误检测和纠正网络，适配中文拼写纠错任务，效果好。
- GPT 模型：本项目基于 PyTorch 实现了用于中文文本纠错的 ChatGLM/LLaMA 模型，模型在中文 CSC 和语法纠错数据集上 finetune，适配中文文本纠错任务，效果好。

2.4.5 文本纠错的评测方法

文本纠错的评测方法或指标可以分为两类：错误检测和错误修正。常用的评测指标有

- Precision（精确率）：纠正正确的错误数 / 纠正的错误总数
- Recall（召回率）：纠正正确的错误数 / 总的错误数
- F1-score（F1 值）：2 * Precision * Recall / (Precision + Recall)

在错误检测任务中，常用的评测指标有：

- True Positive（真正例）：正确检测出的错误数。
- False Positive（假正例）：错误地检测出的错误数。
- False Negative（假负例）：未检测出的错误数。

在错误修正任务中，常用的评测指标有：

- Word Error Rate（词错误率）：纠正的错误单词数 / 总的单词数。
- Position Independent Word Error Rate（位置无关词错误率）：纠正的错误单词数 / 总的单词数，其中，位置无关词错误率不考虑错误单词的位置。

3 文本匹配

文本匹配一直是自然语言处理（NLP）领域一个基础且重要的方向，一般研究两段文本之间的关系。文本相似度计算、自然语言推理、 问答系统、信息检索等，都可以看作针对不同数据和场景的文本匹配应用。这些自然语言处理任务在很大程度上都可以抽象成文本匹配问题， 比如信息检索可以归结为搜索词和文档资源的匹配，问答系统可以归结为问题和候选答案的匹配，复述问题可以归结为两个同义句的匹配。

文本匹配任务数据每一个样本通常由两个文本组成（query，title）。类别形式为 0 或 1，0 表示 query 与 title 不匹配； 1 表示匹配。

文本匹配的常规解决方案，具体如下:

- 基于单塔 Point-wise 范式的语义匹配模型: 模型精度高、计算复杂度高, 适合直接进行语义匹配 2 分类的应用场景。
- 基于单塔 Pair-wise 范式的语义匹配模型: 模型精度高、计算复杂度高, 对文本相似度大小的序关系（ranking）建模能力更强，适合将相似度特征作为上层排序模块输入特征

的应用场景。

- 基于双塔 Point-wise 范式的语义匹配模型：模型计算复杂度更高，适合对延时要求高、根据语义相似度进行粗排的应用场景。

3.1 基于词袋模型的文本匹配

3.1.1 TF-IDF

项目地址：https://github.com/hrs/python-tf-idf

算法描述：

TF-IDF(term frequency–inverse document frequency)是一种用于信息检索与数据挖掘的常用加权技术，常用于挖掘文章中的关键词，而且算法简单高效，常被工业用于最开始的文本数据清洗。TF 指的是是"词频"（Term Frequency，缩写为 TF），另外 IDF 指的是"逆文档频率"（Inverse Document Frequency，缩写为 IDF）。它的主要步骤即是：

（1）计算词频：

$$词频（TF）= \frac{某个词在文章中出现次数}{文章总词数}$$

（2）计算逆文档频率：

$$逆文档频率（IDF）= \log_{10}（\frac{语料库总文档数}{包含该词的文档数+1}）$$

（3）计算 TF-IDF：

$$TF-IDF = 词频(TF) \times 逆文档频率(IDF)$$

3.1.2 BM25

项目地址：https://github.com/dorianbrown/rank_bm25

算法描述：

BM25 是信息检索领域用于计算相关性评分的经典算法，核心是计算 Query（一般是一个句子）和文本集合 D 中每篇文本之间的相关性。我们要对 Query 进行语素解析（一般是分词），在这里以分词为例，我们对 Query 进行分词，得到 q1,q2,.....,qt 这样一个词序列。给定文本 d∈D，现在以计算 Query 和 d 之间的分数（相关性），其表达式如下：

$$Score = \sum_{1}^{t} W_i R(Q_i, d)$$

上面公式中w_i表示权重，也就是 IDF 值（逆文档频率）。$R_{q_i,d}$是分词后的 Query 与文档 d 的相关性得分。其中，BM25 的设计依据一个重要的发现：词频和相关性之间的关系是非

线性的，也就是说，每个词对于文档的相关性分数不会超过一个特定的阈值，当词出现的次数达到一个阈值后，其影响就不在线性增加了，而这个阈值会跟文档本身有关。因此，在刻画单词与文档相似性$R_{q_i,d}$时，BM25 是这样设计的：

$$R（q_i,d）=\frac{(k_1+1)tf_{qd}}{K+tf_{qd}}$$

$$K=k_1(1-b+b*\frac{L_d}{L_{ave}})$$

其中，tf_{qd}是单词 q 在文档 d 中的词频，L_d是文档 d 的长度，L_{ave}是所有文档的平均长度，变量K_1是一个正的参数，用来标准化文章词频的范围。

3.2 基于词嵌入的算法文本匹配

3.2.1 WMD

项目地址：https://github.com/mkusner/wmd

算法描述：WMD（Word Mover's Distance）是一种用于衡量两个文本之间相似性的算法。它基于词嵌入模型，例如 Word2Vec，通过计算将一个文本中的词语转移到另一个文本中所需的最小代价。

它的主要步骤即是：(1) 使用 Word2Vec 或其他词嵌入技术，将文本中的词语映射到向量空间。(2) 利用词嵌入模型，计算每对词语之间的距离，通常使用欧氏距离或余弦相似度。(3) 将词语之间的距离组成一个距离矩阵。(4) 制定一种方法来找到两个文本之间的最佳词语匹配，以最小化词语转移的总代价。这可以通过线性优化问题的求解来实现。(5) 通过将匹配问题的解代入距离矩阵中，计算文档之间的 Word Mover's Distance。

3.2.2 SIF

项目地址：https://github.com/jx00109/sentence2vec

算法描述：SIF（Smooth Inverse Frequency）是一种用于降低文本中常见词对语义相似性的影响的算法。它主要用于生成文本的向量表示，以更好地捕捉文本的语义信息。

SIF 算法步骤：(1) 使用 Word2Vec 或其他词嵌入技术，将文本中的词语映射到实数向量空间。(2) 对于每个词语，计算其逆文档频率（IDF）作为权重，以表示词语的重要性。(3) 对于文本中的每个词语，将其词嵌入向量乘以对应的权重。然后对所有加权后的词嵌入向量求平均，得到文本的向量表示。(4) 使用奇异值分解（SVD）对文本向量进行降维，去除文本中的主成分，以减轻常见词对相似性的影响。(5) 通过去除主成分后的文本向量得到最终的文本表示。

3.3 基于深度学习的文本匹配

3.3.1 SimCSE

项目地址：https://github.com/princeton-nlp/SimCSE

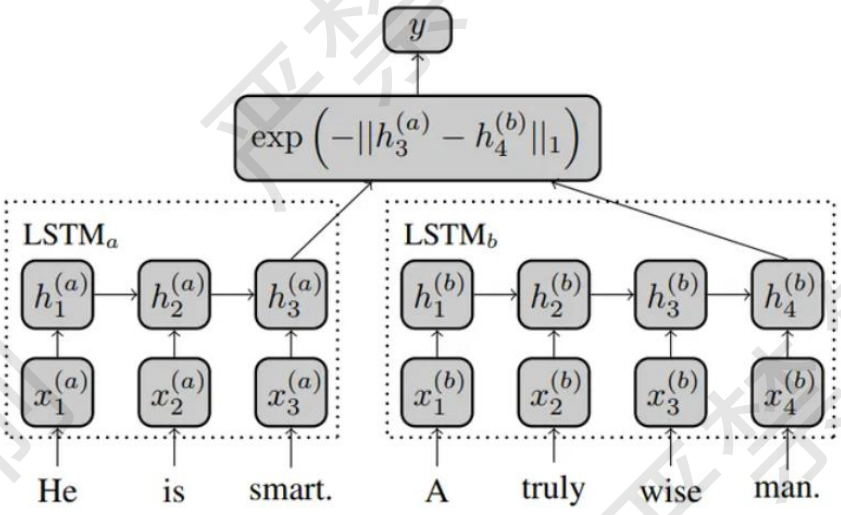
算法描述：SimCSE(Simple Contrastive Learning of Sentence Embeddings)，它通过通过引入对比学习去生成句子级别的向量。该算法分别提出了无监督和有监督的两种方式：首先无监督的情况下，它在神经网络中加入 dropout 来做数据增强来构建正例。因此在一个 batch 中，将同一个句子在模型中的两次输出当作正例，将其他句子的输出全部当作负例。优化对比损失,增加正例之间的相似度，减小负例之间的相似度。在有监督的情况下，通过负采样构造（X,X+,X-）三元组。然后将三元组（X,X+,X-）同时输入到文本向量抽取的模型中进行特征抽取。优化对比损失,增加正例之间的相似度，减小负例之间的相似度。

3.3.2 SiamLSTM

项目地址：

https://github.com/demelin/Sentence-similarity-classifier-for-pyTorch

算法描述：该模型采用两个共享参数的 LSTM(LSMT-a 和 LSTM-b)来捕获需要计算相似度的句子的潜在特征，其中，LSTM 的输入是每个句子的 token 的词向量。每个句子的最终表示为 LSTM 的最后一个时间步的隐藏层输出。可以参考下面的模型图:

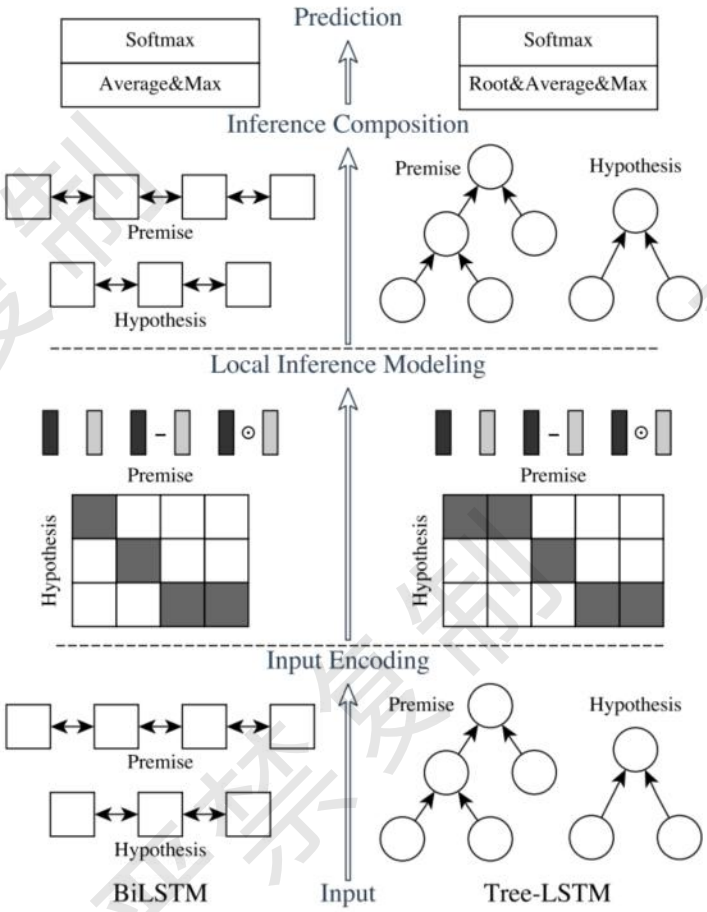


(1) ESIM

项目地址：https://github.com/EternalFeather/ESIM

算法描述：ESIM（Enhanced Sequential Inference Model）模型主要包括三个重要阶段。首先，ESIM 使用 BiLSTM 对两个句子进行编码，即对这两个语句中词语的词向量进行上下文表示。其次，计算注意力权重作为两个句子间词语的相似度，并通过软对齐层（soft alignment layer）得到权重，结合之前的句子编码生成相似性加权后的向量，将两个句子的

向量做差和点积增强局部推理的信息。最后继续使用 BiLSTM 来计算局部推理的信息，并计算平均池和最大池得到更为丰富的多角度向量表示，并连接所有这些向量。可以参考下面的模型图:



(2) DSSM

项目地址：https://github.com/ChrisCN97/DSSM-Pytorch

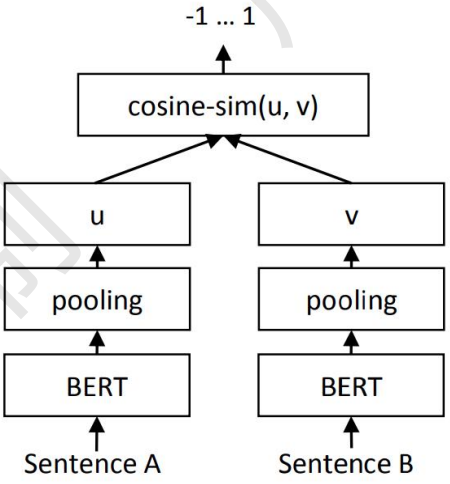
算法描述：DSSM（Deep Structured Semantic Models）总的来说可以分成三层结构，分别是输入层、表示层和匹配层。DSSM 深度语义匹配模型原理就是在训练阶段分别用复杂的深度学习网络构建 query 侧特征的 query embedding 和 doc 侧特征的 doc embedding，线上 infer 时通过计算两个语义向量的 cos 距离来表示语义相似度，最终获得语义相似模型。这个模型既可以获得语句的低维语义向量表达 sentence embedding，还可以预测两句话的语义相似度。

(3) SBERT

项目地址：https://github.com/UKPLab/sentence-transformers

算法描述：SBERT（Sentence-BERT）沿用了孪生网络的结构和在 NLP 任务上有着卓越表现的 BERT，两个 Sentence Encoder 使用的是同一个 BERT，并在其后加入了一个池化

（pooling）操作来实现输出相同大小的句向量。最后通过计算输出向量的余弦相似度来经计算两个句子的文本相似性。

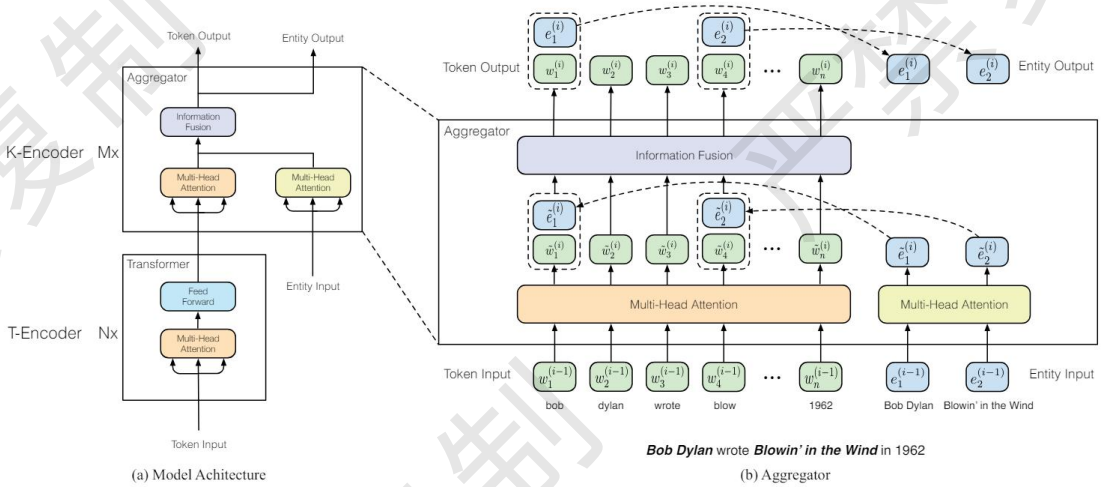


3.4 基于知识图谱的文本匹配

3.4.1 ESIM

项目地址：https://github.com/thunlp/ERNIE

算法描述：BERT 本身是一种基于大语料库的无监督预训练模型，为了让其学习到更多结构化的知识，许多前沿工作开始研究如何把外部知识注入到 BERT 当中。由清华大学发布的 ERNIE-thu 使用维基百科语料库进行预训练，并使用语料中的 Anchor Link 来获取实体，通过 Wikidata 训练出的 TransE 向量作为实体的特征。TransE 是知识图谱嵌入（Knowledge Graph Embedding）方法的一种，和词嵌入类似，这些方法的核心是把知识图谱中的所有实体和关系映射到连续的向量空间当中，由此让机器能够更好地理解知识图谱的结构化信息。在 BERT-base 的基础上，ERNIE-thu 保留了 6 层的 Encoder，但是将原本的 Decoder 改成了文中提出的 K-Encoder 来学习知识图谱中的实体信息。



(a) Model Achitecture　　(b) Aggregator

数据集：

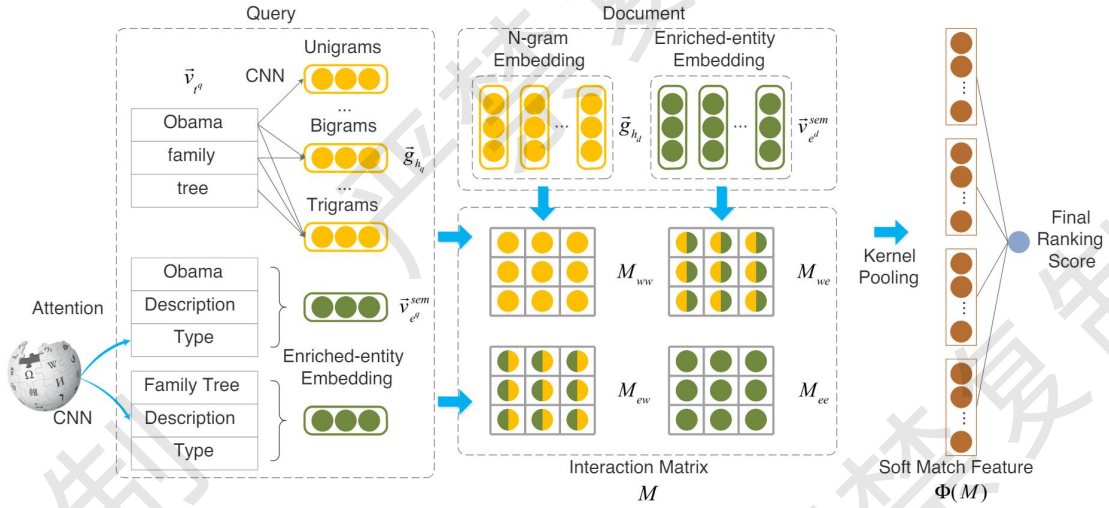
wikidata：https://www.wikidata.org/wiki/Wikidata:Database_download

维基数据是一个自由开放的知识库，人和机器都可以阅读和编辑。维基数据充当其维基媒体姊妹项目（包括维基百科、维基导游、维基词典、维基文库等） 结构化数据的中央存储。维基数据还为维基媒体项目之外的许多其他网站和服务提供支持。维基数据的内容可以在免费许可下使用，使用标准格式导出，并且可以与链接数据网络上的其他开放数据集互连。

3.4.2 EDRM

项目地址：https://github.com/thunlp/EntityDuetNeuralRanking

算法描述：EDRM 采用了 CN-DBpedia 构建的知识图谱（Knowledge Graph，KG）作为外部知识的来源。CN-DBpedia 是一个由百度百科、互动百科、中文维基百科构建的大型中文知识库，其格式为（subject, relation, object）形式的三元组，其中包含 10,341,196 个实体 以及 88,454,264 种关系。文中提出的模型 EDRM 将 IR 任务中的文本对通过词+实体的特征来表示。对于一段文本中出现的实体，作者构造三种特征来进行表示：Entity Embedding，Description Embedding，Type Embedding 来获得 Enriched-entity Embedding。



4 OCR 识别

4.1 通用文字识别

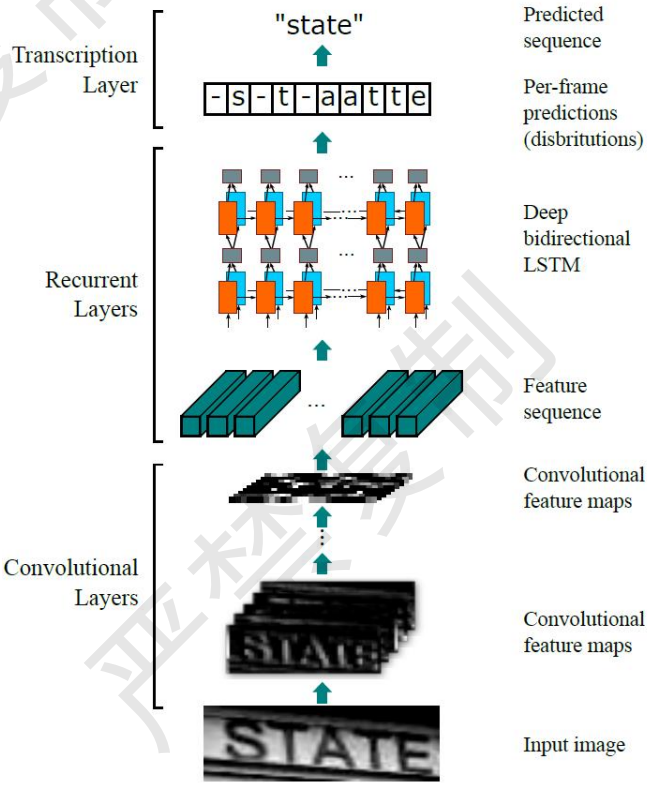
4.1.1 CRNN+CTC

（1）算法描述

CRNN (Convolutional Recurrent Neural Network) 专门用于处理图像中的序列化文字

识别问题。它结合了卷积神经网络（CNN）和循环神经网络（RNN），利用这两种网络的优点来提高文字识别的准确性和效率。该算法认为文字识别是对序列的预测方法，所以采用了对序列预测的 RNN 网络。通过 CNN 将图片的特征提取出来后采用 RNN 对序列进行预测，最后通过一个 CTC 的转录层得到最终结果。

模型分为三个部分：卷积层、循环层、转录层。转录层的作用是将前面通过 CNN 层和 RNN 层得到的预测序列转换成标记序列，得到最终的识别结果。CRNN 算法无需字符分割，并且可以处理不定长文本，结构灵活简单。模型结构如图所示。



（2）项目地址

百度飞桨 Paddle 库：https://github.com/PaddlePaddle/PaddleOCR

高 star 复现（pytorch）：https://github.com/meijieru/crnn.pytorch

（3）数据集：SVT

街景文本 （SVT） 数据集是从 Google 街景中收集的。此数据中的图像文本具有很高的可变性，并且通常具有较低的分辨率。在处理室外街道图像时，我们注意到两个特征。（1）图像文本通常来自商业标牌，（2）通过地理商业搜索很容易获得企业名称。这些因素使 SVT 集特别适合在野外发现单词：给定街景图像，目标是识别附近企业的单词。

数据集地址：https://github.com/open-mmlab/mmocr/tree/main/dataset_zoo/svt

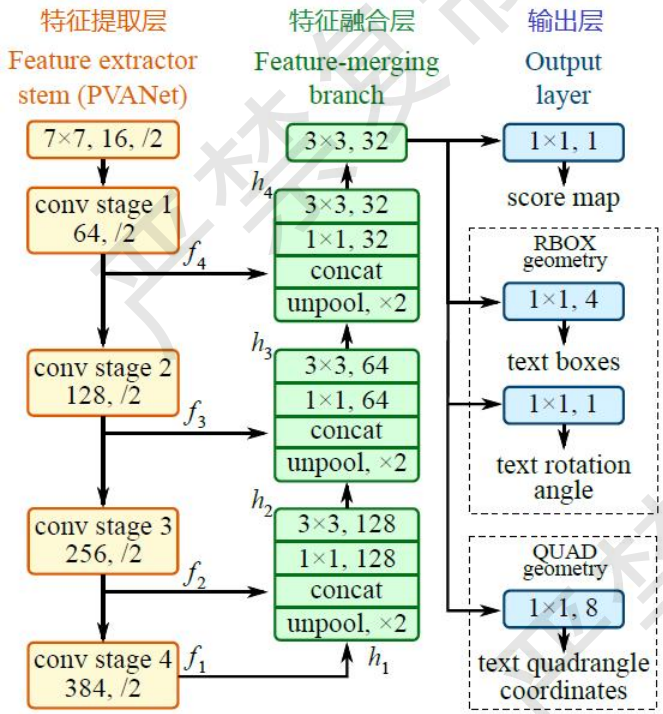
4.1.2 EAST

（1）算法描述

EAST (Efficient and Accurate Scene Text Detector) 是一个高效且准确的场景文字检测模型，它在 2017 年被提出，用于直接从全图中预测文本块的位置和形状。

EAST 模型设计用于实时场景文字检测，其处理速度快，可以在不牺牲准确性的情况下运行在标准硬件上有以下特点：(1). EAST 模型能够快速且精确地处理复杂背景下的场景文本。与传统的文字检测模型（如基于候选区域提取的方法）不同，EAST 直接从图像中预测文字的位置，无需先生成文字候选区域。这种直接预测方法减少了计算步骤，提高了处理速度。(2).可以检测不同尺度和形状的文本，包括倾斜的和弯曲的文本。这是通过在模型中引入可旋转的边界框实现的。(3).EAST 模型是一个基于全卷积网络的单阶段文字检测器，能够输出文字的分数图、几何形状等信息。模型使用卷积层处理整个图像，生成像素级的预测，这包括文本存在的概率和文本区域的几何参数。

其结构如下图所示，它由特征提取层（如 VGG16、ResNet 等）、特征融合层、输出层组成。



（2）算法地址

百度飞桨 Paddle 库：https://github.com/PaddlePaddle/PaddleOCR

高 star 复现（tensorflow）：https://github.com/argman/EAST

（3）数据集

①COCO-Text

COCO-Text 数据集是用于文本检测和识别的数据集。它基于 MS COCO 数据集，该数据集包含复杂的日常场景图像。COCO-Text 数据集包含非文本图像、清晰文本图像和难以辨认的文本图像。总共有 22184 个训练图像和 7026 个验证图像，其中至少有一个可读文本实例。

数据集地址：https://datasets.activeloop.ai/docs/ml/datasets/coco-text-dataset/

②ICDAR 2013

ICDAR 2013 数据集侧重于从原生数字图片中提取文本内容，例如在线和通过电子邮件使用的图片（原生数字图像是为在线传输而创建的媒体文件）。ICDAR 2013 数据集由 462 张照片组成，其中 229 张用于训练集，233 张用于测试集。文本本地化、文本分割和单词识别都是与从原生数字图片中进行文本提取相关的挑战。

数据集地址：https://datasets.activeloop.ai/docs/ml/datasets/icdar-2013-dataset/

4.1.3 TextBoxes++

（1）算法描述

TextBoxes++ 是一个专门设计来检测和识别场景中文本的深度学习模型，是 TextBoxes 模型的扩展版。TextBoxes 初版主要是用于识别水平排列的文本行，而 TextBoxes++ 对此进行了改进，使其能够有效地检测和识别多方向文本，包括水平、垂直和倾斜文本。这使得 TextBoxes++ 成为处理自然场景中各种文本排列的理想选择。

TextBoxes++ 的核心特征：

多方向文本检测。TextBoxes++ 通过改进原有的 TextBoxes 模型，增加了对倾斜和垂直文本的检测能力。这一点是通过采用倾斜的边界框来实现的，边界框可以以任意角度存在，从而覆盖各种文本方向。

基于 SSD 框架。TextBoxes++ 基于单阶段检测器 SSD（Single Shot MultiBox Detector）。它在 SSD 的基础上修改和优化了预测层，专门用于文本检测任务。

适应文本行的感受野。模型使用了长矩形的锚框（anchor boxes），这些锚框比传统的正方形或矩形锚框更适合文本的形状。

TextBoxes++ 使用 VGG16 作为特征提取器。

（2）算法地址

https://github.com/MhLiao/TextBoxes_plusplus

（3）数据集

① COCO-Text: https://datasets.activeloop.ai/docs/ml/datasets/coco-text-dataset/

② ICDAR 2015: https://rrc.cvc.uab.es/?ch=4&com=downloads

4.1.4 Star-net

（1）算法描述：

STAR-Net (Spatial Attention Residue Network) 是一个专为场景文本识别设计的深度学习框架。它通过结合空间注意力机制和残差网络，有效地处理了自然场景中的文本识别问题，尤其是在复杂背景和不规则文本排列的情况下。STAR-Net 的设计目标是提高模型对文本特征的聚焦能力，从而提高识别的准确性和鲁棒性。

STAR-Net 由特征提取网络、空间转换网络（STN）、空间注意力机制、序列建模和识别层四个部分组成。

特征提取网络使用残差网络（ResNet）作为基础的特征提取器。

空间转换网络（STN）允许网络自动学习到图像的最优空间变换，从而校正图像中的文本，使其更有利于后续的识别任务。STN 可以处理图像的缩放、旋转和倾斜等问题。

空间注意力模块用于增强特征图中与目标任务（文本识别）相关的区域，同时抑制不相关的背景信息。这种注意力机制帮助模型集中资源处理图像中的关键部分（即文本区域）。

序列建模部分通常使用双向 LSTM 来处理从注意力模块传出的特征，以捕捉文本行内的序列依赖性。使用连接时序分类（CTC）损失函数来进行端到端的文本识别。CTC 支持不定长的序列输出，无需对字符间的精确位置进行预定义。

（2）算法地址

百度飞浆 Paddle 库：https://github.com/PaddlePaddle/PaddleOCR

（3）数据集

① ICDAR 2013：https://datasets.activeloop.ai/docs/ml/datasets/icdar-2013-dataset/

② SVT：https://github.com/open-mmlab/mmocr/tree/main/dataset_zoo/svt

4.1.5 OFA-OCR

（1）算法描述

OFA-OCR 是一种将多模态预训练模型转换为文本识别的方法。具体来说，该算法将文本识别重新转换为图像标题，并直接将统一的视觉语言预训练模型传输到最终任务。无需对大规模注释或合成文本识别数据进行预训练。其中，OFA 是一个统一的序列到序列预训练模型（支持英文和中文），它统一了模态（即跨模态、视觉、语言）和任务。

（2）算法地址

OFA 官方库（具体为 OFA-OCR）：https://github.com/ofa-sys/ofa

（3）数据集

benchmarking-chinese-text-recognition（中文数据集）

该数据集包括场景数据集、Web 数据集、文档数据集、手写数据集等。并包括了一些基线方法。是一个包含了中文文字识别各种数据的仓库。

数据集地址：https://github.com/FudanVI/benchmarking-chinese-text-recognition

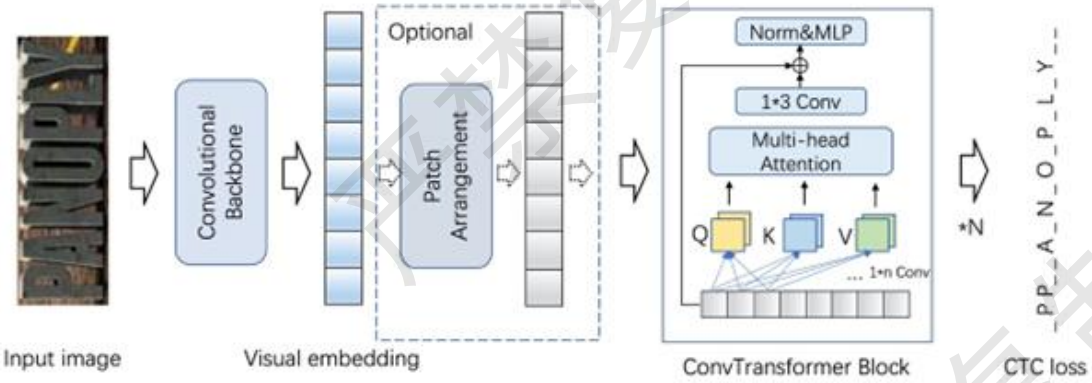
4.2 印刷文字识别

4.2.1. 读光文字识别

项目地址：

https://www.modelscope.cn/models/iic/cv_convnextTiny_ocr-recognition-document_damo/summary

算法描述：文字识别，即给定一张文本图片，识别出图中所含文字并输出对应字符串。本模型主要包括三个主要部分，Convolutional Backbone 提取图像视觉特征，ConvTransformer Blocks 用于对视觉特征进行上下文建模，最后连接 CTC loss 进行识别解码以及网络梯度优化。识别模型结构如下图所示。



数据集：MTWI 网络文本图像识别公开数据集。

https://www.modelscope.cn/datasets/iic/WebText_Dataset/summary

4.2.2. OFA 文字识别

模型描述：OFA(One-For-All)是通用多模态预训练模型，使用简单的序列到序列的学习框架统一模态（跨模态、视觉、语言等模态）和任务（如图片生成、视觉定位、图片描述、图片分类、文本生成等）。

下载地址：

https://www.modelscope.cn/models/iic/ofa_ocr-recognition_document_base_zh/summary

评估结果：OFA 在文字识别（ocr recognize）在公开数据集(including RCTW, ReCTS, LSVT, ArT, CTW)中进行评测，在准确率指标上达到 SOTA 结果，具体如下：

Model	Scene	Web	Document	Handwriting	Avg
SAR	62.5	54.3	93.8	31.4	67.3
TransOCR	63.3	62.3	96.9	53.4	72.8
MaskOCR-base	73.9	74.8	99.3	63.7	80.8
OFA-OCR	82.9	81.7	99.1	69.0	86.0

数据集：Benchmarking-Chinese-Text-Recognition：

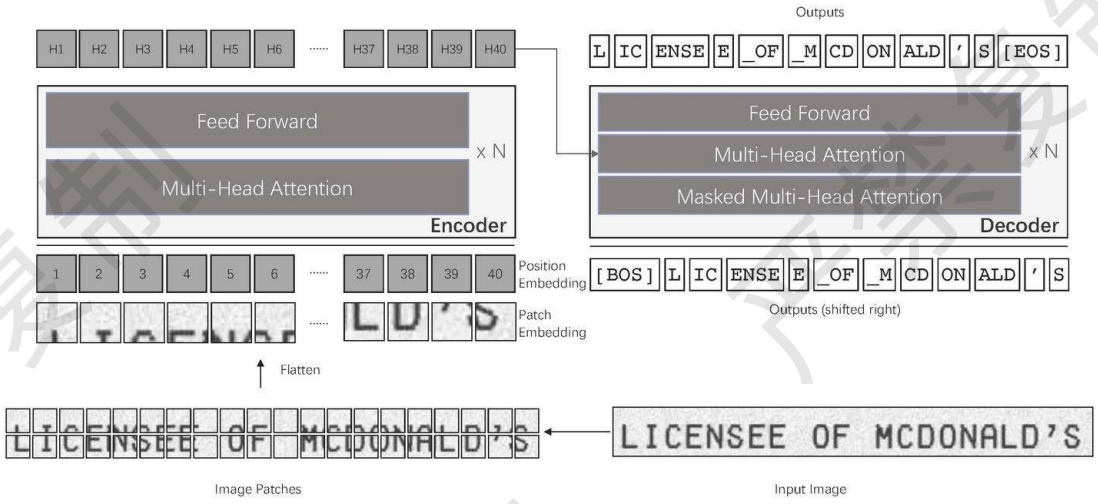
https://github.com/FudanVI/benchmarking-chinese-text-recognition

4.3 手写文字识别

4.3.1TrOCR

项目地址：https://github.com/chineseocr/trocr-chinese

算法描述：TrOCR（Transformer OCR）是由 Microsoft 发布的一个 OCR 识别模型，该模型基于 Transformer 模型，编码器由预训练的 Vision Transformer 组成，主要采用了 DeIT 作为 Vision transformer 模型。解码器由预训练的 language transformer 模型组成，主要采用 RoBERTa 与 UniLM 模型。



首先，图像被分解成小块，并添加相应的位置编码；第二步，将图像输入到 TrOCR 模

型，经过图像编码器，编码器主要包括多头注意力机制与 feed forward 前馈神经网络；第三步经过解码器部分，解码器的输入是标准的文本，其文本需要跟编码器的数据进行注意力机制的计算；最后，对编码输出进行解码以获得图像中的文本。

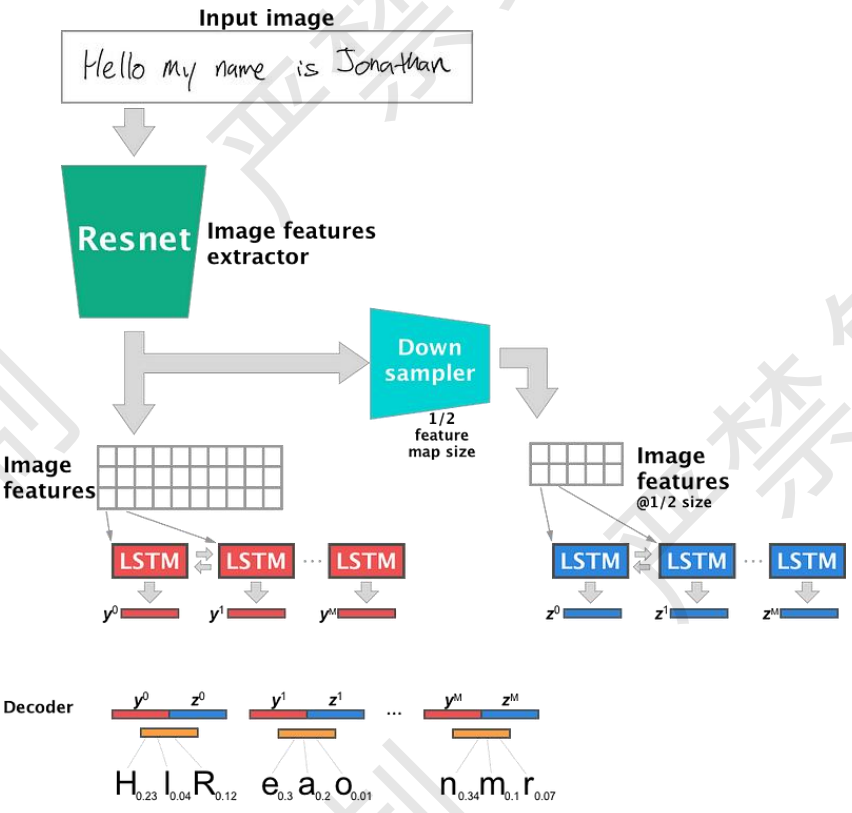
4.3.2 Apache MXNet

项目地址：

https://github.com/awslabs/handwritten-text-recognition-for-apache-mxnet

算法描述：Apache MXNet 是由 Amazon 工程师 Jonathan Chung 开发的手写文字识别模型，该模型输入采用包含一行文字的图像，并返回一个包含每个字符出现概率的矩阵。具体来说，矩阵的大小为序列长度×字符词汇。

该模型的核心是利用 CNN 提取图像特征，并将特征输入双向 LSTM，训练网络以优化连接时态分类(CTC)损失。直观地说，CNN 生成的图像特征在空间上与输入图像对齐，然后，图像特征沿文本方向切分，并依次输入 LSTM。这个网络被称为 CNN-biLSTM，其相较于多维 LSTM 的计算成本更低。该模型还为图像特征提供了多个下采样来帮助识别大小不同的手写文本图像（例如，只包含 1 个单词的行与包含 7 个单词的行）。图像特征提取器使用的是预先训练好的 Res-net。最后，CNN-biLSTM 的输出被输入解码器，以预测图像每个垂直切片上字符的概率分布。



4.4 图片文字识别

4.4.1PP-OCR

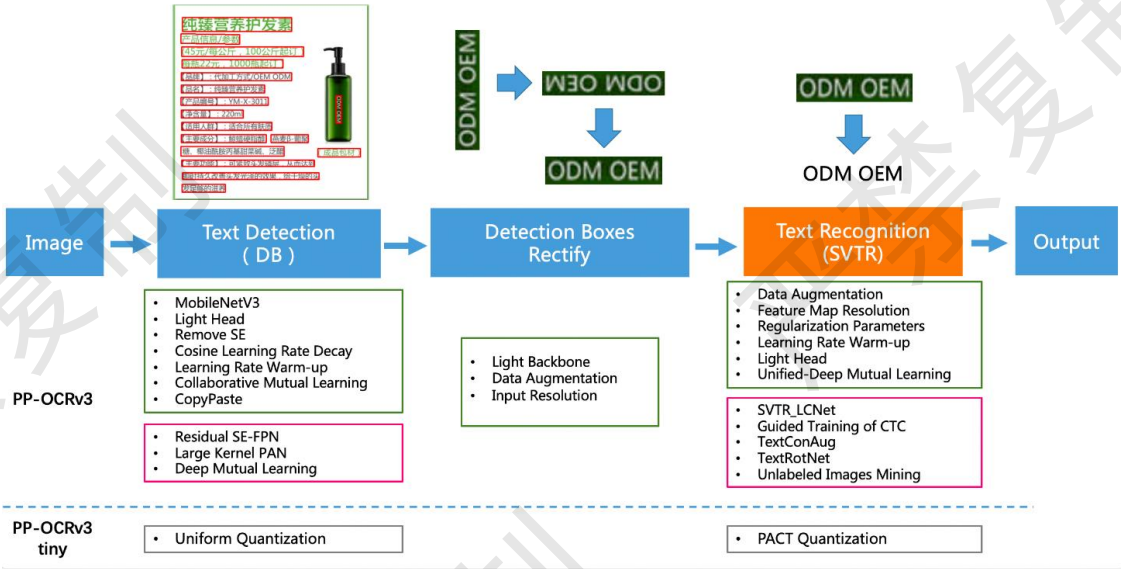
项目地址：https://github.com/PaddlePaddle/PaddleOCR

算法描述：

PaddleOCR 是一个基于飞桨开发的 OCR 系统，其技术体系包括文字检测、文字识别、文本方向检测和图像处理等模块，具有高精度、高效易用、多语种支持等特点。其中，PP-OCR 是 PaddleOCR 自研的实用的超轻量 OCR 系统。在实现前沿算法的基础上，考虑精度与速度的平衡，进行模型瘦身和深度优化，使其尽可能满足产业落地需求。

PP-OCR 是一个两阶段的 OCR 系统，其中文本检测算法选用 DB，文本识别算法选用 CRNN，并在检测和识别模块之间添加文本方向分类器，以应对不同方向的文本识别。PP-OCRv3 在 PP-OCR 的基础上，针对检测模型和识别模型，进行了共计 9 个方面的升级：

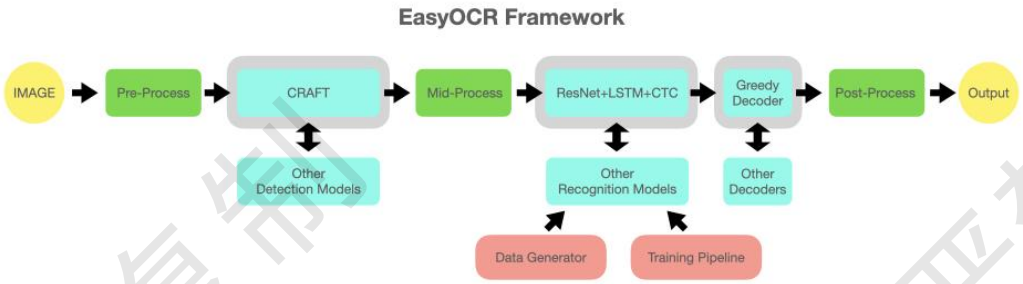
PP-OCRv3 检测模型对 CML 协同互学习文本检测蒸馏策略进行了升级，分别针对教师模型和学生模型进行进一步效果优化。其中，在对教师模型优化时，提出了大感受野的 PAN 结构 LK-PAN 和引入了 DML 蒸馏策略；在对学生模型优化时，提出了残差注意力机制的 FPN 结构 RSE-FPN。PP-OCRv3 的识别模块是基于文本识别算法 SVTR 优化。SVTR 不再采用 RNN 结构，通过引入 Transformers 结构更加有效地挖掘文本行图像的上下文信息，从而提升文本识别能力。PP-OCRv3 通过轻量级文本识别网络 SVTR_LCNet、Attention 损失指导 CTC 损失训练策略、挖掘文字上下文信息的数据增广策略 TextConAug、TextRotNet 自监督预训练模型、UDML 联合互学习策略、UIM 无标注数据挖掘方案，6 个方面进行模型加速和效果提升。



4.4.2EasyOCR

项目地址：https://github.com/JaidedAI/EasyOCR

算法描述：EasyOCR 是一个使用 Java 语言，基于 CRNN 实现的 OCR 识别引擎，借助几个简单的 API，即能使用 Java 语言完成图片内容识别工作。

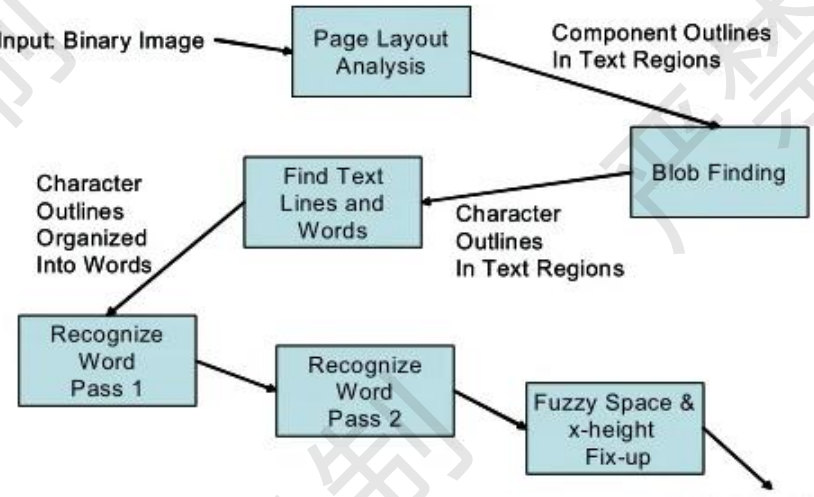


EasyOCR 的检测执行算法使用的是 CRAFT 算法，它是一种基于分割的算法，无需进行大量候选框的回归，也无需进行 NMS 后处理，因此极大提升了速度，并且它是字符级别的文本检测器，定位的是字符，对于尺寸缩放不敏感，无需多尺度训练和预测来解决尺度方差问题；EasyOCR 的识别模型使用的是 CRNN，主要由三个组件组成：特征提取（Resnet）、序列标记（LSTM）以及解码器（CTC），用于识别执行的训练管道使用的是 deep-text-recognition-benchmark 的改进版本，分别从矫正、特征提取、BiLSTM 和 Attention 这四个模块进行了一定的改进，提高了对文本识别的精度，减少了耗时。

4.4.3TesseractOCR

项目地址：https://github.com/tesseract-ocr/tesseract

算法描述：Tesseract OCR 是一个非常经典的开源 OCR 引擎，最初由 Hewlett-Packard 开发，现在由 Google 维护。它以准确性和多功能性闻名，可以提取数据并将扫描的文档、图像和手写文字转换为机器理解的文本。支持 100 多种语言，并兼容多种操作系统，并且提供了非常方便的命令行界面。Tesseract 支持 unicode（UTF-8），可以识别超过 100 种语言；支持多种图像格式，包括 PNG、JPEG 等；支持多种输出格式，包括纯文本、PDF 等。



Tesseract 的主要识别步骤如下：

1.连通区域分析,检测出字符区域区域(轮廓外形)，以及子轮廓。在此阶段轮廓线集成为块区域。

2.由字符轮廓和块区域得出文本行，以及通过空格识别出单词。固定字宽文本通过字符单元分割出单个字符，而对百分号的文本(Proportional text)通过一定的间隔和模糊间隔(fuzzy spaces)来分割。

3.依次对每个单词进行分析，采用自适应分类器，分类器有学习能力，先分析且满足条件的单词也作为训练样本，所以后面的字符(比如页尾)识别更准确:此时，页首的字符识别比较不准确，所以 Tesseract 会再次对识别不太好的字符识别是其精度得到提高。

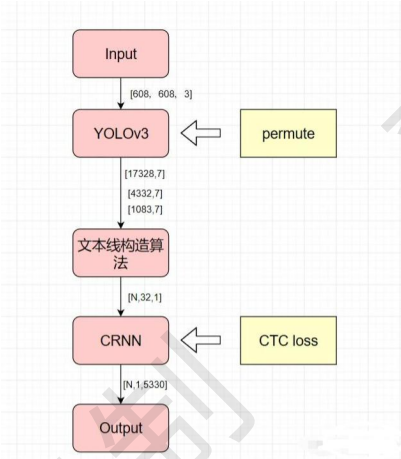
4.最后，识别含糊不清的空格，及用其他方法，如由笔画高度(x-height)，识别小写字母(small-cap)的文本。

4.5 发票识别

ChineseOCR（基于 YOLO3 和 CRNN 实现）

项目地址：https://github.com/guanshuicheng/invoice

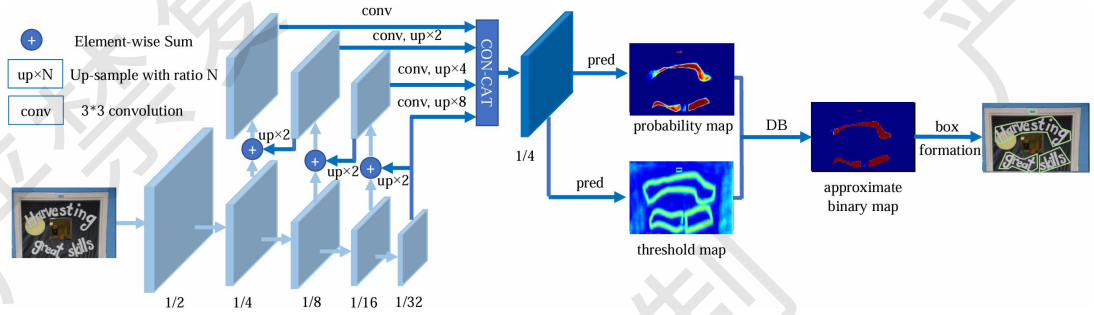
算法描述：ChineseOCR 是一个基于 Python 的 OCR 工具，其核心是利用 TensorFlow 和 Keras 开发的深度学习模型。该项目的目标是简化中文文字识别的过程，让开发者和普通用户都能方便地利用这一技术。ChineseOCR 使用了 YOLOv3，其是一个多尺度的通用目标检测网络，总共有三个尺度上的输出，共有 252 层，没有全连接层。YOLOv3 是基于 Darknet 框架实现的。Darknet 是一个基于 C 和 CUDA 的一个开源深度学习框架，特点是轻量级，没有任何依赖项。ChineseOCR 使用了 CRNN（Convolutional Recurrent Neural Network）架构，这是一种结合卷积神经网络（CNN）和循环神经网络（RNN）的模型，专门用于序列数据的建模，如文字识别。此外，CTC（Connectionist Temporal Classification）损失函数也被应用在训练过程中，允许模型处理不同长度的输入序列，无需预先对齐训练样本。



4.6 营业执照识别

4.6.1 DBNet 文本检测

算法描述：DBNet 是基于分割的文本检测算法，算法将可微分二值化模块(Differentiable Binarization)引入了分割模型，使得模型能够通过自适应的阈值图进行二值化，并且自适应阈值图可以计算损失，能够在模型训练过程中起到辅助效果优化的效果。经过验证，该方案不仅提升了文本检测的效果而且简化了后处理过程。相较于其他文本检测模型，DBNet 在效果和性能上都有比较大的优势，是当前常用的文本检测算法。



算法地址：

https://github.com/MhLiao/DB

https://github.com/PaddlePaddle/PaddleOCR

数据集：ICDAR 2015 https://rrc.cvc.uab.es/?ch=4&com=downloads

4.6.2 SVTR 文本识别

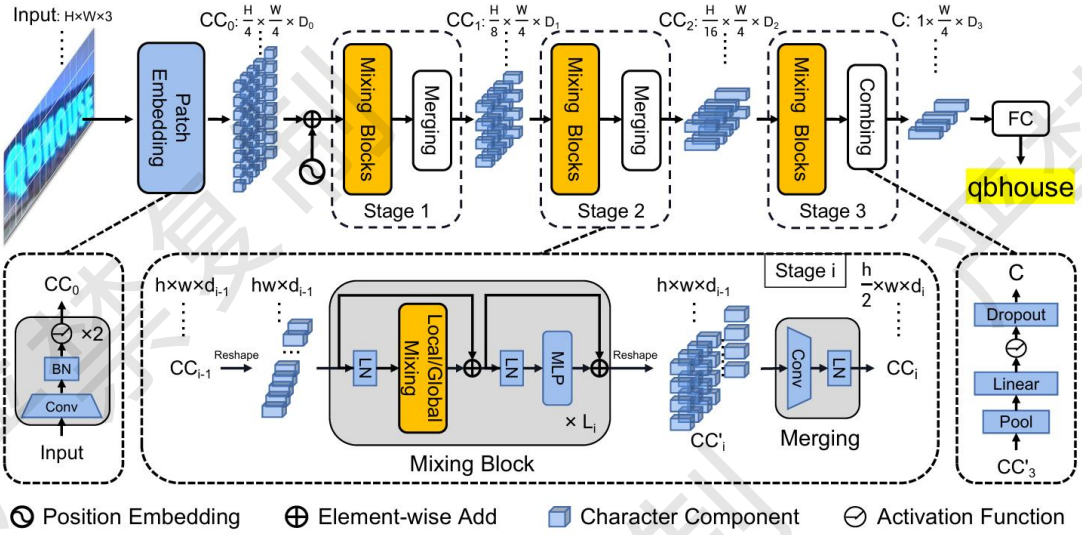
算法描述：主流的场景文本识别模型通常包含两个模块：用于特征提取的视觉模型和用于文本转录的序列模型。这种架构虽然准确，但复杂且效率较低，限制了在实际场景中的应用。SVTR 提出了一种用于场景文本识别的单视觉模型，该模型在 patch-wise image tokenization 框架内，完全摒弃了序列建模，在精度具有竞争力的前提下，模型参数量更少，速度更快，SVTR 首先将图像文本分解为名为字符组件的小块，然后通过组件级混合、合并和/或组合反复执行分层阶段。设计了全局和局部混合块来感知角色间和角色内模式，从而产生多粒度的角色组件感知。因此，字符可以通过简单的线性预测来识别。

SVTR 主要有以下几点贡献：

1）首次发现单视觉模型可以达到与视觉语言模型相媲美甚至更高的准确率，并且其具有效率高和适应多语言的优点，在实际应用中很有前景。

2）SVTR 从字符组件的角度出发，逐渐的合并字符组件，自下而上地完成字符的识别。

3）SVTR 引入了局部和全局 Mixing，分别用于提取字符组件特征和字符间依赖关系，与多尺度的特征一起，形成多粒度特征描述。



算法地址：

https://github.com/open-mmlab/mmocr/tree/main/configs/textrecog/svtr

https://github.com/PaddlePaddle/PaddleOCR

数据集：

https://github.com/fudanvi/benchmarking-chinese-text-recognition

ICDAR 2015 https://rrc.cvc.uab.es/?ch=4&com=downloads

4.7 办公文档识别

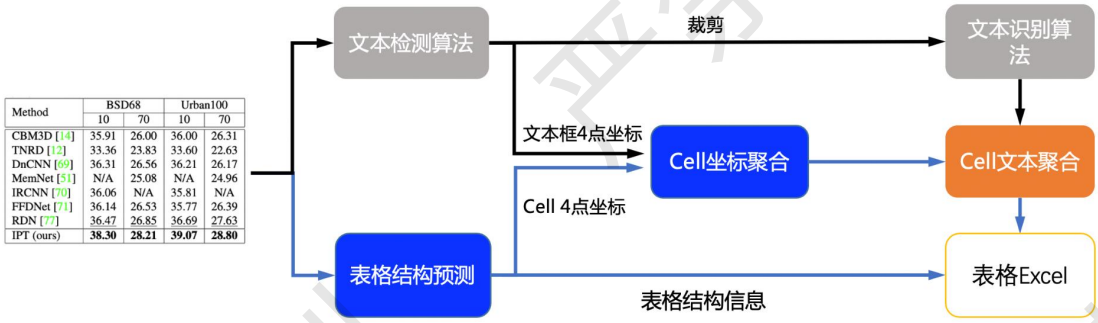
4.7.1 表格文字识别

算法描述：

表格识别主要包含三个模型：1)单行文本检测-DB，2)单行文本识别-CRNN，3）表格结构和 cell 坐标预测-SLANet。

主要流程为：首先图片由单行文字检测模型检测到单行文字的坐标，然后送入识别模型拿到识别结果。然后图片由 SLANet 模型拿到表格的结构信息和单元格的坐标信息。由单行文字的坐标、识别结果和单元格的坐标一起组合出单元格的识别结果。最后单元格的识别结果和表格结构一起构造表格的 html 字符串。

Method	BSD68		Urban100	
	10	70	10	70
CBM3D [14]	35.91	26.00	36.00	26.31
TNRD [12]	33.36	23.83	33.60	22.63
DnCNN [69]	36.31	26.56	36.21	26.17
MemNet [51]	N/A	25.08	N/A	24.96
IRCNN [70]	36.06	N/A	35.81	N/A
FFDNet [71]	36.14	26.53	35.77	26.39
RDN [77]	36.47	26.85	36.69	27.63
IPT (ours)	**38.30**	**28.21**	**39.07**	**28.80**



项目地址：https://github.com/PaddlePaddle/PaddleOCR/tree/main/ppstructure/table

数据集：

PubTabNet 数据集 https://github.com/wangwen-whu/WTW-Dataset

WTW 中文场景表格数据集 https://ai.100tal.com/dataset

4.7.2 文档版面分析

模型描述：版面分析指的是对图片形式的文档进行区域划分，定位其中的关键区域，如文字、标题、表格、图片等。版面分析算法基于 PP-ORC 模型库中 PaddleDetection 的轻量模型 PP-PicoDet 进行开发，包含英文、中文、表格版面分析 3 类模型。其中，英文模型支持 Text、Title、Tale、Figure、List5 类区域的检测，中文模型支持 Text、Title、Figure、Figure caption、Table、Table caption、Header、Footer、Reference、Equation10 类区域的检测，表格版面分析支持 Table 区域的检测。

项目地址：https://github.com/PaddlePaddle/PaddleOCR/tree/main/ppstructure/layout

数据集：

publaynet 数据集：https://github.com/ibm-aur-nlp/PubLayNet

CDLA 数据集：https://github.com/buptlihang/CDLA

4.7.3 印章检测识别

模型描述：PP-OCR 平台中通过使用 PaddleDetection 工具库和 PaddleOCR 实现印章检测和印章文字识别任务。其中 PaddleDetection 用于实现印章检测，PaddleOCR 用于实现文字识别。任务过程中首先使用 PaddleLable 工具标注印章检测数据，标注内容包括印章的位置以及印章中文字的位置和文字内容。模型采用 mobilenetv3 为 backbone 的 ppyolo 算法完成印章检测任务。识别任务中有两种方式，一种是两阶段的 OCR 算法，另一种为端对端的 OCR 算法。两阶段中使用 DB++和 SVTR 进行训练，端对端使用 PGNet 算法。

项目地址：https://github.com/PaddlePaddle/PaddleOCR/tree/dygraph/applications

5 命名实体识别

5.1 命名实体识别语料库

(1) CoNLL-2003

CoNLL-2003 是一个经典的英文命名实体识别（NER）语料库，被广泛用于训练和评估 NER 模型。它由 CoNLL-2003 共享任务组织发布，主要包含了新闻文章和维基百科的文本。CoNLL-2003 中的每个单词都被标记为四种实体类型之一：人名、地名、组织名和未知实体。此外，还标注了如缩写、数字、标点等其他类型的信息，以帮助更准确地识别实体。CoNLL-2003 中的数据集分为三个部分：训练集（14041 句）、开发集（3250 句）和测试集（3453 句）。

除了标注实体类型外，CoNLL-2003 还提供了一些其他的信息，例如词性标注、句法结构分析等，以帮助进行更深入的分析和建模。在实际应用中，CoNLL-2003 通常被用作 NER 模型的基准数据集，并与其他模型进行比较和评估。

(2) OntoNotes

OntoNotes 是一个广泛使用的英文语料库，旨在支持多领域的自然语言处理任务。它提供了各种类型的文本数据，包括新闻文章、对话、采访等，涵盖了多个语言领域和文本风格。OntoNotes 的主要目标是提供丰富的语义信息和语言学注释，以支持诸如命名实体识别（NER）、词义消歧、指代消解等任务。除了 NER 外，OntoNotes 还提供了其他级别的实体标注，如人物、组织、地点、时间、数字等。

OntoNotes 的设计基于语义角色标注（Semantic Role Labeling）理论，试图捕捉句子中的语义信息和语义角色关系。因此，在 OntoNotes 中，不仅标注了实体，还标注了动词的论元（如施事者、受事者、工具等）和语义角色。

(3) GENIA

GENIA 是一个专门用于生物医学领域的英文语料库，特别适用于基因和蛋白质实体的识别。它提供了大量的科学论文摘要和全文数据，并对其中的生物医学实体进行了标注。GENIA 语料库的目标是支持生物信息学和生物医学自然语言处理的研究。它涵盖了多个生物医学子领域，如基因组学、蛋白质组学、生物化学等。在 GENIA 中，主要关注的实体类型是基因和蛋白质，但也包括其他相关的实体类型，如细胞、化合物等。

GENIA 语料库的数据来源于 PubMed 等权威的生物医学文献数据库。为了进行实体标注，GENIA 通过人工标注和自动标注相结合的方式进行。标注人员根据预先定义的标注规范，对文本中的实体进行标注，并进行质量控制。除了实体标注外，GENIA 还提供了一些其他的注释信息，如基因和蛋白质之间的关系、实体的属性等。这些附加信息可以帮助进一

步理解和分析生物医学文本中的信息。

5.2 通用命名实体识别方法

5.2.1 基于标记的命名实体识别

项目地址：meizhiju/layered-bilstm-crf (github.com)

方法描述：

Tag-based 命名实体识别方法通常使用标记来指示文本中命名实体的边界和类别。这种方法将文本中的每个词或子词与一个标记相关联，以表示该词是否属于某种命名实体类型，例如人名、地名、组织名等。

一种常见的 tag-based 方法是基于序列标注的方法，其中最常见的方法是命名实体识别中的 BIO 标记法（Beginning, Inside, Outside）。在 BIO 标记法中，每个词被标记为 B、I 或 O：

B（Beginning）：表示该词是一个命名实体的开始部分。

I（Inside）：表示该词是一个命名实体的中间部分。

O（Outside）：表示该词不是命名实体的一部分。

通过这种方式，可以构建一个序列标注模型（如条件随机场、循环神经网络、Transformer 等），该模型能够在标记序列上学习命名实体的边界和类型。

5.2.2 基于跨度的命名实体识别

项目地址：ljynlp/W2NER: Source code for AAAI 2022 paper: Unified Named Entity Recognition as Word-Word Relation Classification (github.com)

方法描述：

span-based 命名实体识别方法是一种基于跨度（span）的方法，通过识别文本中的实体跨度来确定命名实体的边界和类型。与 tag-based 方法不同，span-based 方法直接识别并标注命名实体的跨度，而不是单独标记每个词。

在 span-based 方法中，通常使用标注标签来表示文本中的命名实体跨度，这些标签通常是实体类型的缩写，例如 PER 表示人名、LOC 表示地名、ORG 表示组织名等。每个标签与文本中的一个跨度相对应，该跨度被视为相应类型的命名实体。

span-based 方法通常使用序列标注模型（如条件随机场、循环神经网络、Transformer 等）来预测文本中的命名实体跨度。模型通过在输入文本中识别实体跨度的开始和结束位置，从而确定每个命名实体的边界和类型。

相较于 tag-based 方法，span-based 方法在处理具有复杂结构或交叉重叠实体的文本时更具优势，因为它直接处理实体跨度而不需要显式地标记每个词。但是，span-based 方法可

能需要更大的训练数据和更复杂的模型来学习实体跨度的边界和类型。

5.2.3 基于生成的命名实体识别

项目地址：universal-ie/UIE: Unified Structure Generation for Universal Information Extraction (github.com)

方法描述：

基于生成的命名实体识别方法通过将 NER 任务视为序列生成任务来处理，即将输入文本序列映射到对应的命名实体标签序列。在训练阶段，模型学习从输入文本中识别和生成命名实体标签的模式，而在推断阶段，模型根据上下文信息生成命名实体标签序列。生成型 NER 方法通常使用深度学习模型，如循环神经网络（RNN）、长短期记忆网络（LSTM）、变压器模型等，通过大量标记的文本数据进行训练。尽管生成型 NER 方法能够处理各种类型的命名实体和复杂的文本结构，但也存在一些挑战，如模型训练需要大量数据和计算资源、生成效率较低以及模型输出需要后处理等。

5.3 命名实体识别工具

5.3.1 StanfordNER

官网：Named Entity Recognition - CoreNLP (stanfordnlp.github.io)

Stanford Named Entity Recognizer（StanfordNER）是由斯坦福大学自然语言处理组开发的一种基于统计学习的命名实体识别工具。它能够识别文本中的命名实体，并将它们标记为不同的类别，如人名、地名、组织名等。StanfordNER 基于条件随机场（Conditional Random Fields, CRF）模型，结合了词性标注、词性转换和词形还原等特征，通过训练数据学习到模型参数，从而在给定文本中自动识别和标记命名实体。它在文本处理、信息抽取和语义分析等领域广泛应用，为实体识别任务提供了强大的支持。

5.3.2 MALLET

官网：Mallet: MAchine Learning for LanguagE Toolkit | Mallet (mimno.github.io)

MALLET（MAchine Learning for LanguagE Toolkit）是一个由麻省大学 Amherst 分校的计算机科学家开发的开源机器学习工具包，用于自然语言处理和文本挖掘任务。它提供了一系列用于分类、聚类、序列标注等任务的算法和工具，并且支持多种机器学习模型，包括隐马尔可夫模型（Hidden Markov Models, HMM）、条件随机场（Conditional Random Fields, CRF）、最大熵模型（Maximum Entropy Models, MaxEnt）等。MALLET 具有简单易用的接口和丰富的功能，可以用于文本分类、命名实体识别、文档聚类、情感分析等各种文本分析任务，并且在学术界和工业界得到广泛应用

5.3.3 Hanlp

官网：HanLP

Hanlp（Han Language Processing）是由一系列自然语言处理工具组成的开源项目，旨在为中文文本处理提供高效、准确的解决方案。它包含了丰富的功能模块，包括分词、词性标注、命名实体识别、依存句法分析、语义角色标注等。Hanlp 采用了基于深度学习的先进技术，在各项任务上都取得了优异的性能表现，并且支持多种预训练模型，用户可以根据自己的需求选择合适的模型进行文本处理。Hanlp 的设计简洁灵活，提供了友好的接口和丰富的文档，深受学术界和工业界的欢迎和广泛应用。

工具	简介	访问地址
Stanford NER	斯坦福大学开发的基于条件随机场的命名实体识别系统，该系统参数是基于CoNLL、MUC-6、MUC-7和ACE命名实体语料训练出来的。	官网 | GitHub地址
MALLET	麻省大学开发的一个统计自然语言处理的开源包，其序列标注工具的应用中能够实现命名实体识别。	官网
Hanlp	HanLP是一系列模型与算法组成的NLP工具包，由大快搜索主导并完全开源，目标是普及自然语言处理在生产环境中的应用。支持命名实体识别。	官网 | GitHub地址
NLTK	NLTK是一个高效的Python构建的平台,用来处理人类自然语言数据。	官网 | GitHub地址
SpaCy	工业级的自然语言处理工具，遗憾的是不支持中文。	官网 | GitHub地址
Crfsuite	可以载入自己的数据集去训练CRF实体识别模型。	文档 | GitHub地址

5.4 垂直领域命名实体识别方法

5.4.1 医疗领域命名实体识别

1、RaNER

项目地址：

RaNER 命名实体识别-中文-医疗领域-base · 模型库 (modelscope.cn)

在医疗领域，命名实体识别（NER）任务旨在从医学文本中识别和标记具有特定语义含义的实体，包括但不限于疾病名称、药物、治疗方法、医学过程和解剖结构等。这项任务对于自然语言处理在医学信息提取方面的应用至关重要，能够帮助医疗从业者和研究人员从大量的医学文本中快速准确地获取关键信息。医疗 NER 任务的挑战在于医学术语的多样性和复杂性，医学文本的语言结构复杂，缩写、同义词和术语变体等现象普遍存在，因此需要结合领域知识和高度专业化的算法模型来解决这些问题。

当前的研究主要集中在深度学习技术来构建医学 NER 系统，以实现对医学文本中各种实体的准确识别和标记。

5.4.2 新闻领域

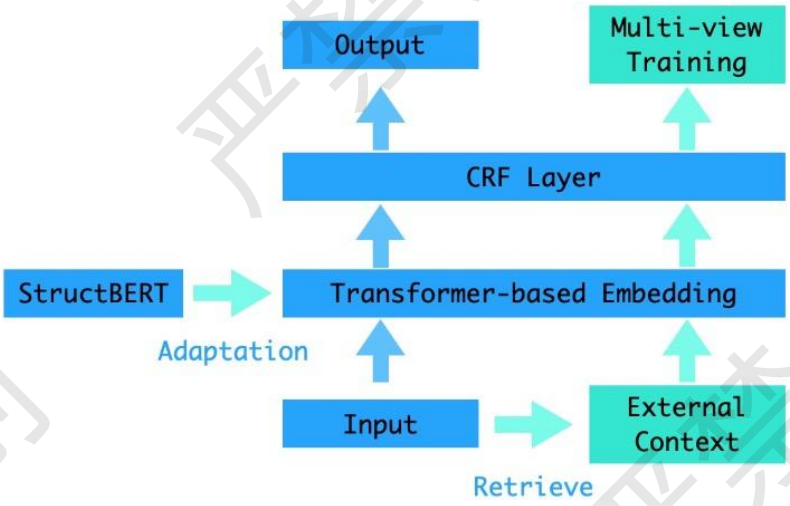
项目地址：

RaNER 命名实体识别-中文-新闻领域-base · 模型库 (modelscope.cn)

新闻领域的命名实体识别（NER）任务旨在从新闻文本中准确识别和标记具有特定语义含义的实体，如人名、地名、组织机构名等。在新闻报道中，这些命名实体往往扮演着重要角色，对于理解新闻内容、提取关键信息以及进行信息检索具有重要意义。然而，由于新闻文本的多样性和复杂性，以及命名实体的语义变化和表达方式的多样性，新闻 NER 任务面临着诸多挑战。

为了解决这些挑战，通常采用基于深度学习的方法构建新闻 NER 系统。这些系统能够从大规模的新闻文本中准确识别和标记各种命名实体，为新闻分析、信息抽取和知识发现提供有力支持。

本方法采用 Transformer-CRF 模型，使用 StructBERT 作为预训练模型底座，结合使用外部工具召回的相关句子作为额外上下文，使用 Multi-view Training 方式进行训练。 模型结构如下图所示：



5.4.3 电商领域

项目地址：

RaNER 命名实体识别-中文-电商领域-base · 模型库 (modelscope.cn)

电商领域的命名实体识别（NER）任务旨在从电商平台上的文本数据中准确识别和标记具有特定语义含义的实体，如产品名称、品牌、价格、规格等。这些命名实体在电商平台上

扮演着重要角色，对于用户搜索、推荐系统、广告投放等环节至关重要。然而，由于电商平台的多样性和复杂性，以及产品信息的丰富多变，电商 NER 任务面临着诸多挑战。

为了解决这些挑战，通常采用基于机器学习和深度学习的方法，如序列标注模型、迁移学习和注意力机制等，来构建高效的电商 NER 系统。这些系统能够从大规模的电商数据中准确识别和标记各种命名实体，为用户提供个性化的购物体验，提高电商平台的服务质量和用户满意度。

严禁复制