自然语言处理基础

CS2916 大语言模型

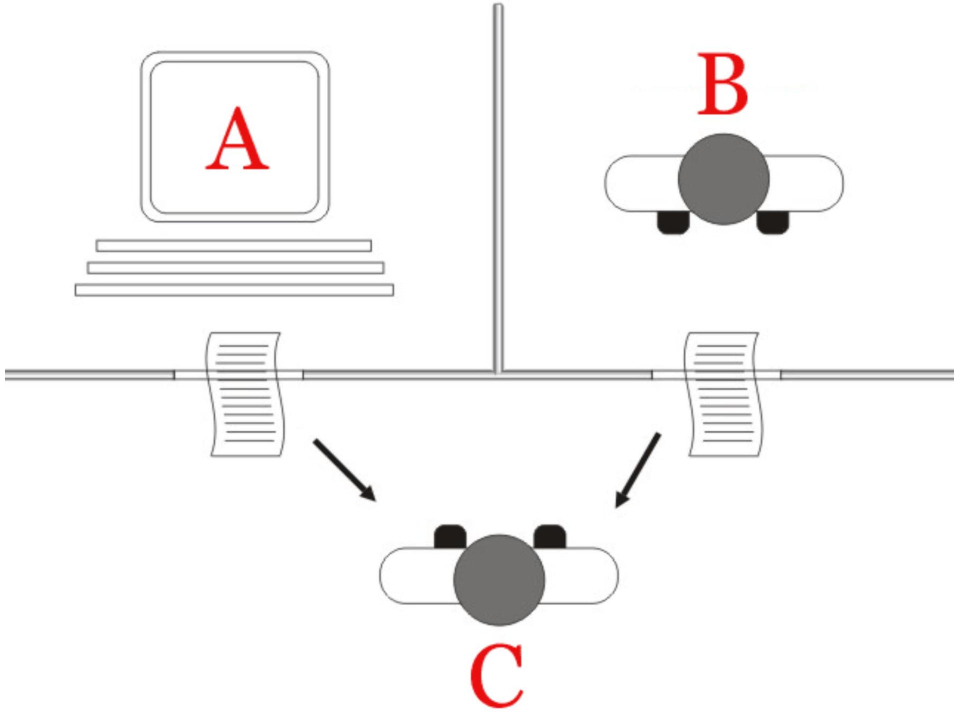
飲水思源 愛國榮校

https://plms.ai/teaching/index.html

图灵测试 (1950)

本质上是一个评估问题，解决方法：Reference-based

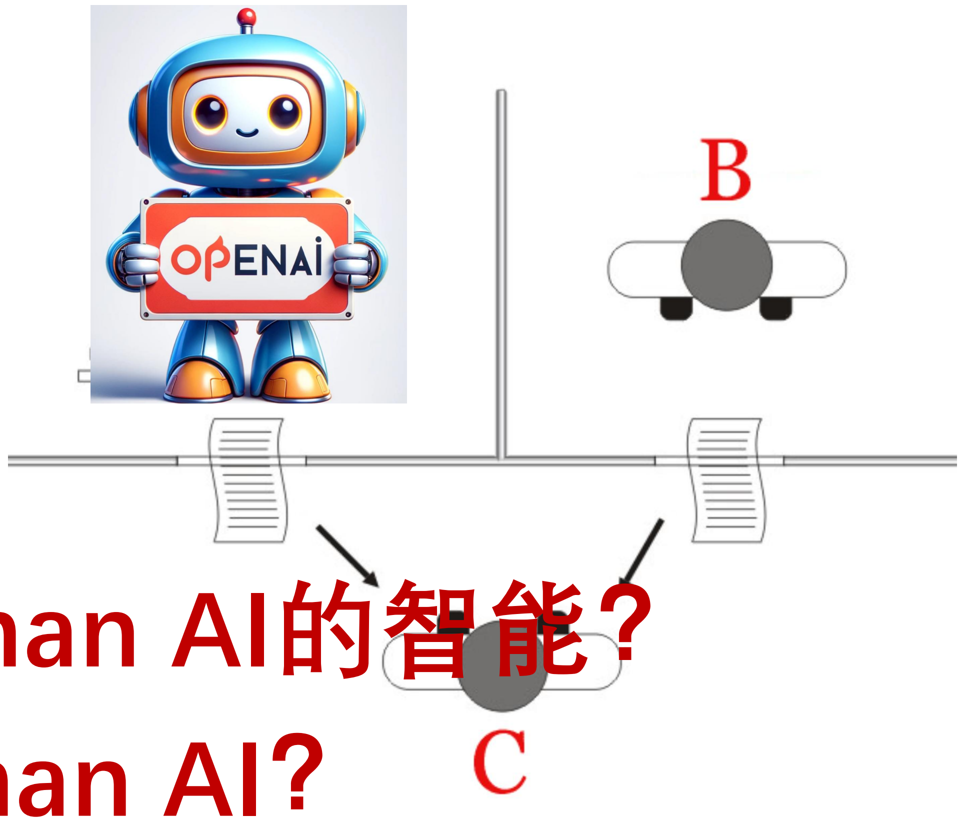
- □ 目的
  - 检验机器的行为是否类似于人类的智能行为
- □ 测试方法
  - 能否以人类无法区分的方式思考或表达思考
- □ 涉及到的技术
  - 自然语言处理、自动推理、计算机视觉、机器人学等



图灵测试 (1950)

本质上是一个评估问题，解决方法：
Reference-based

- □ 目的
  - ■ 检验机器的行为是否类似于人类的智能行为
- □ 测试方法
  - ■ 能否以人类无法区分的方式思考或表达思考
- □ 涉及到的技术
  - ■ 自然语言处理、自动推理、计算机视觉、机器人学等



思考：如何评估superhuman AI的智能？

如何训练superhuman AI?

Measuring Progress on Scalable Oversight for Large Language Models, Bowman et al.2022
Scalable Meta-Evaluation of LLMs as Evaluators via Agent Debate, Chern et al.2024
Superalignment: https://openai.com/blog/introducing-superalignment
Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision, OpenAI 2024

什么是自然语言处理?

语言学家
刘涌泉

自然语言处理是**人工智能**领域的主要内容，即利用电子计算机 等工具对人类所特有的语言信息（包括口语信息和文字信息）进行各种 加工，并建立各种类型的人-机-人系统。自然语言理解是其核心，其中 包括语音和语符的自动识别以及语音的自动合成。”

Natural language processing (NLP) is an **interdisciplinary** subfield of **computer science** and **linguistics**. It is primarily concerned with giving computers the ability to **support and manipulate human language**.

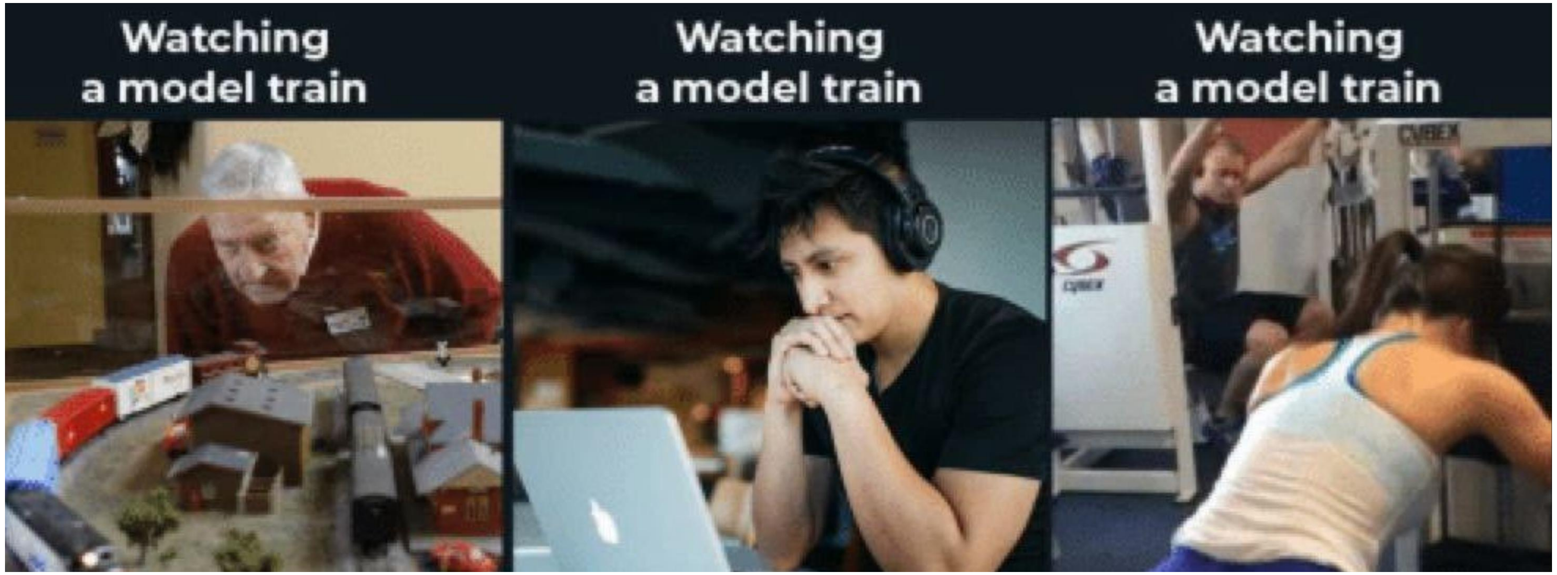
维基百科

宗成庆
老师

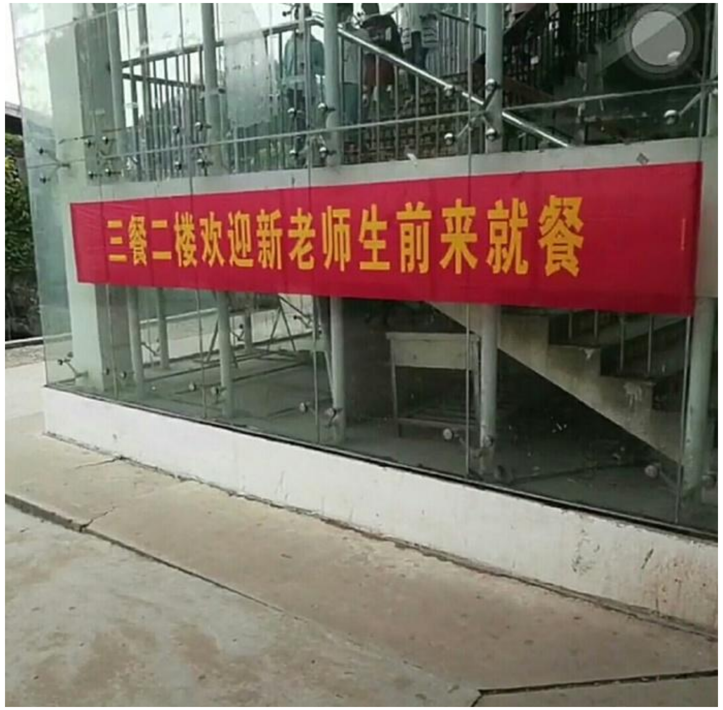
我们从事自然语言理解研究的任务也就是研究和探索针对**具体应用**目的的**新方法和新技术**， 使实现系统的性能表现尽量**符合人类理解的标准和要求**。

自然语言处理挑战

多义性



结构复杂性



递归性

“早上点早点早上早点早点吃”

自然语言处理相关书籍/课程

□ 相关书籍

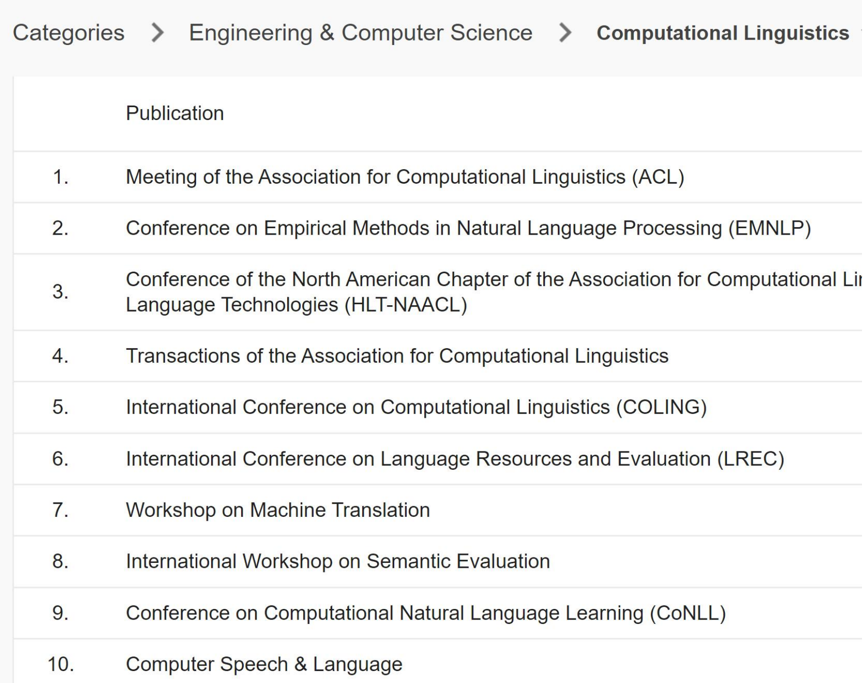
- **Foundations of Statistical Natural Language Processing** Christopher Manning and Hinrich Schütze (1999)
- **An Introduction to Natural Language Processing**, Daniel Jurafsky and James Martin (2008)
- **Neural Network Methods for Natural Language Processing**，Yoav Goldberg
- 自然语言处理综论，宗成庆老师

□ 相关课程

- **交大:** 赵海、俞凯、陈露、林洲汉老师开设了自然语言处理课程
- **Stanford：** CS224n
- **CMU:** CS11-747

自然语言处理相关学术会议

- □ ACL
  - 成立于1962年，每年一次
  - NLP和计算语言学**最高级别**的会议
  - 在北美、欧洲、亚洲分年会
- □ EMNLP
  - 发起于1996年，专注于NLP技术的经验方法
  - 随着统计方法和机器学习技术应用广泛而兴起
- □ NAACL
- □ TACL （期刊)
  - 每个月1号都可以投稿
  - 审稿周期和ACL相当
- □ COLING
  - 成立于1965年，两年一次



Categories > Engineering & Computer Science > **Computational Linguistics**

	Publication
1.	Meeting of the Association for Computational Linguistics (ACL)
2.	Conference on Empirical Methods in Natural Language Processing (EMNLP)
3.	Conference of the North American Chapter of the Association for Computational Li Language Technologies (HLT-NAACL)
4.	Transactions of the Association for Computational Linguistics
5.	International Conference on Computational Linguistics (COLING)
6.	International Conference on Language Resources and Evaluation (LREC)
7.	Workshop on Machine Translation
8.	International Workshop on Semantic Evaluation
9.	Conference on Computational Natural Language Learning (CoNLL)
10.	Computer Speech & Language

NLP领域影响力最大的会议（期刊）Top10

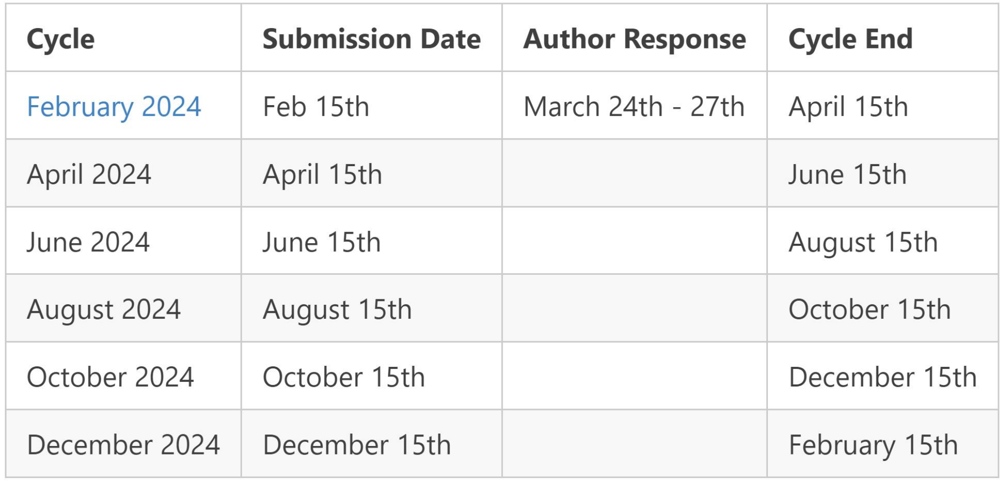
根据谷歌学术指标：
https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_computationallinguistics

自然语言处理相关学术会议

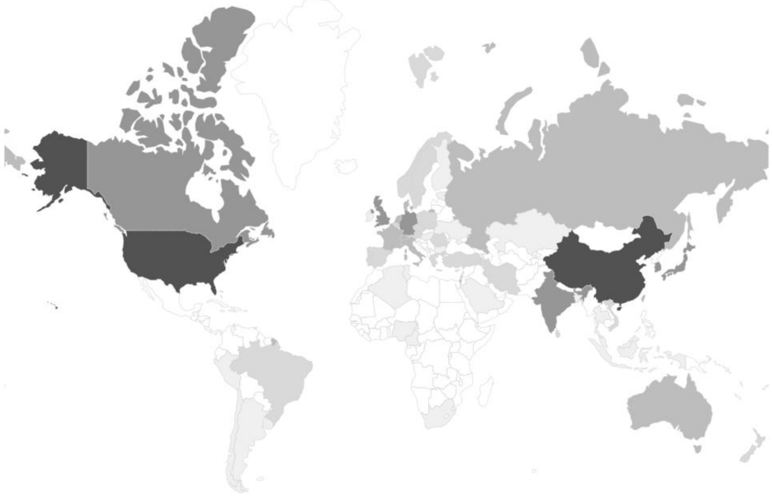
- 其它国际会议：
  - AAAI/IJCAI
  - ICLR/NeurIPS/ICML
- 国内会议
  - CCL
    - 创建于1991年
    - 中国中文信息学会的旗舰会议
  - NLPCC
    - 国际自然语言处理与计算会议

自然语言处理投稿机制

- □ ACL Rolling Review (ARR)
  - ■ 背景：整个AI领域发展迅猛，投稿激增，一年一次的会议投稿审稿时间太长，无法满足技术发展与更新迭代
  - ■ 发起人：CMU教授Graham Neubig率先提议ARR，两阶段
    - ● 集中滚动评审
    - ● 提交投稿至特定会议



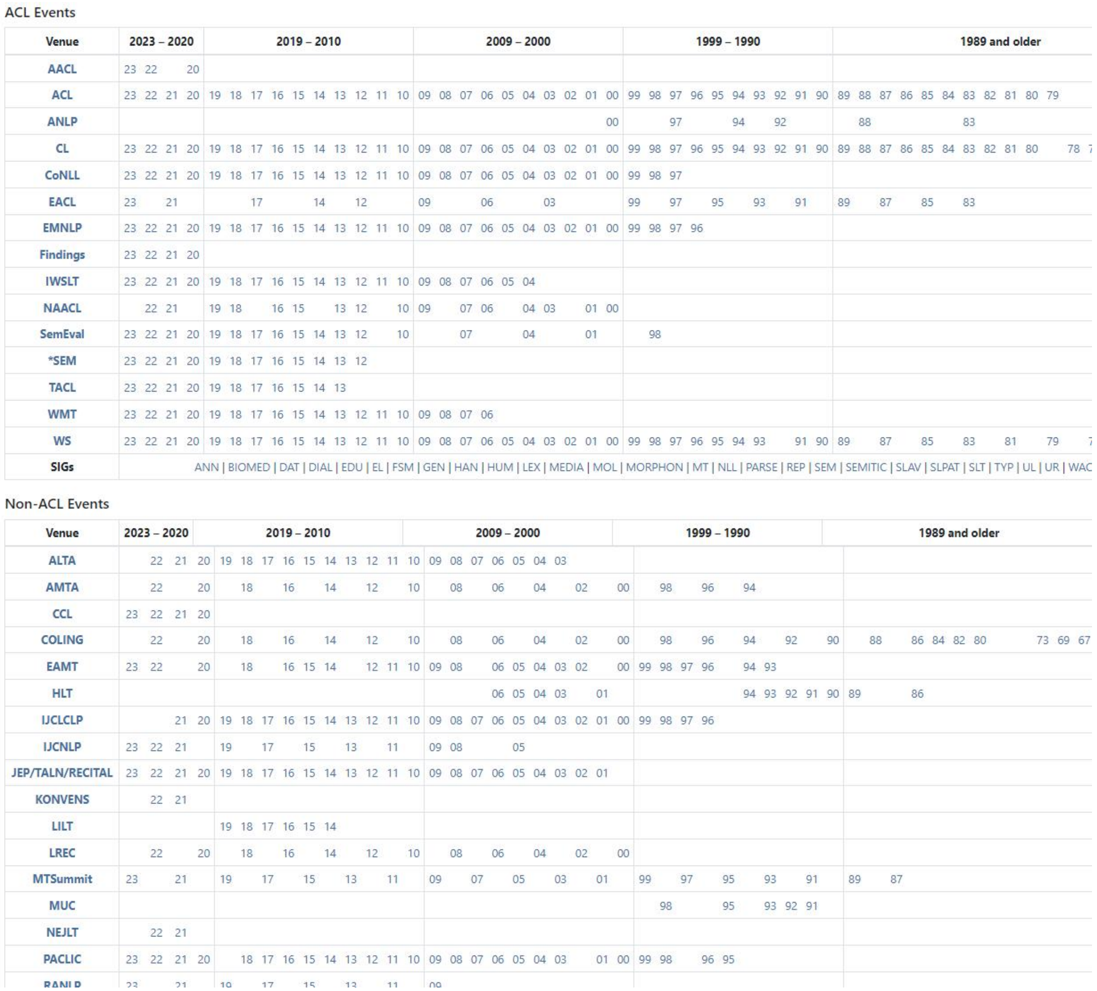
Cycle	Submission Date	Author Response	Cycle End
February 2024	Feb 15th	March 24th - 27th	April 15th
April 2024	April 15th		June 15th
June 2024	June 15th		August 15th
August 2024	August 15th		October 15th
October 2024	October 15th		December 15th
December 2024	December 15th		February 15th



https://stats.aclrollingreview.org/

自然语言处理论文集

□ ACL Anthology

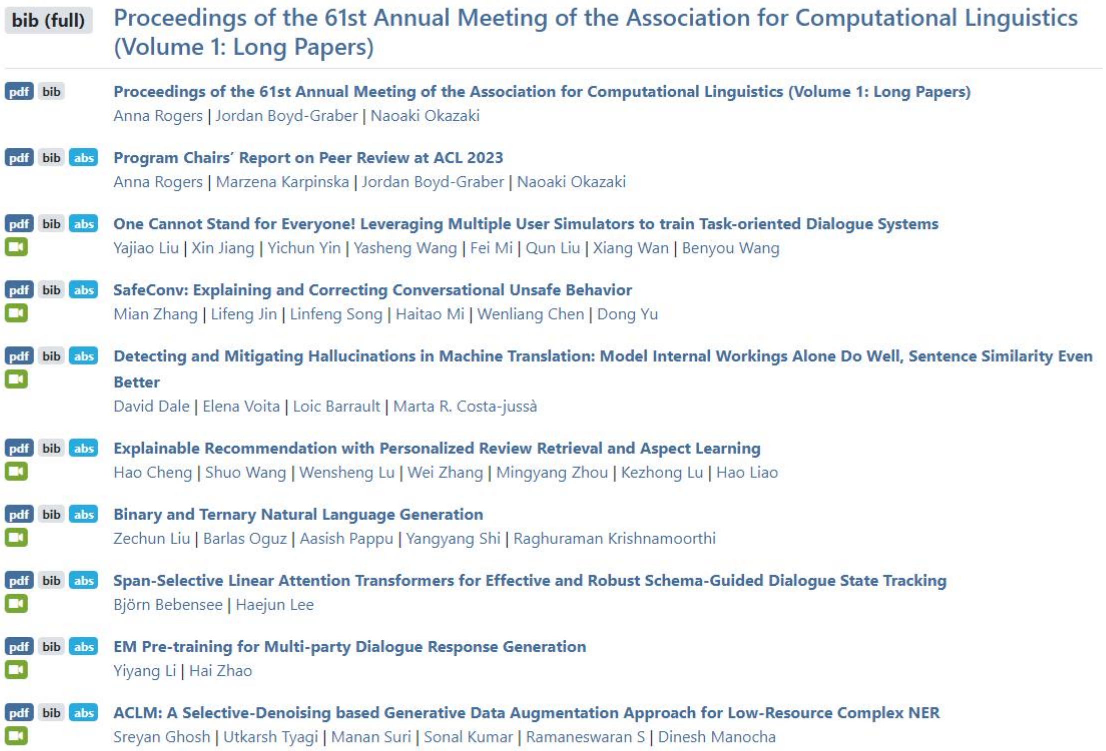


ACL Events

Venue	2023 – 2020	2019 – 2010	2009 – 2000	1999 – 1990	1989 and older
AACL	23 22 20				
ACL	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 92 91 90	89 88 87 86 85 84 83 82 81 80 79
ANLP			00	97 94 92	88 83
CL	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 92 91 90	89 88 87 86 85 84 83 82 81 80 78 7
CoNLL	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97	
EACL	23 21	17 14 12	09 06 03	99 97 95 93 91	89 87 85 83
EMNLP	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96	
Findings	23 22 21 20				
IWSLT	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04		
NAACL	22 21	19 18 16 15 13 12 10	09 07 06 04 03 01 00		
SemEval	23 22 21 20	19 18 17 16 15 14 13 12 10	07 04 01	98	
*SEM	23 22 21 20	19 18 17 16 15 14 13 12			
TACL	23 22 21 20	19 18 17 16 15 14 13			
WMT	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06		
WS	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96 95 94 93 91 90	89 87 85 83 81 79 7
SIGs	ANN | BIOMED | DAT | DIAL | EDU | EL | FSM | GEN | HAN | HUM | LEX | MEDIA | MOL | MORPHON | MT | NLL | PARSE | REP | SEM | SEMITIC | SLAV | SLPAT | SLT | TYP | UL | UR | WAC				

Non-ACL Events

Venue	2023 – 2020	2019 – 2010	2009 – 2000	1999 – 1990	1989 and older
ALTA	22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03		
AMTA	22 20	18 16 14 12 10	08 06 04 02 00	98 96 94	
CCL	23 22 21 20				
COLING	22 20	18 16 14 12 10	08 06 04 02 00	98 96 94 92 90	88 86 84 82 80 73 69 67
EAMT	23 22 20	18 16 15 14 12 11 10	09 08 06 05 04 03 02 00	99 98 97 96 94 93	
HLT			06 05 04 03 01	94 93 92 91 90	89 86
IJCLCLP	21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01 00	99 98 97 96	
IJCNLP	23 22 21	19 17 15 13 11	09 08 05		
JEP/TALN/RECITAL	23 22 21 20	19 18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 02 01		
KONVENS	22 21				
LILT		19 18 17 16 15 14			
LREC	22 20	18 16 14 12 10	08 06 04 02 00		
MTSummit	23 21	19 17 15 13 11	09 07 05 03 01	99 97 95 93 91	89 87
MUC				98 95 93 92 91	
NEJLT	22 21				
PACLIC	23 22 21 20	18 17 16 15 14 13 12 11 10	09 08 07 06 05 04 03 01 00	99 98 96 95	
RANLP	23 21	19 17 15 13 11	09		



bib (full) **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**

pdf bib **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**
Anna Rogers | Jordan Boyd-Graber | Naoaki Okazaki

pdf bib abs **Program Chairs’ Report on Peer Review at ACL 2023**
Anna Rogers | Marzena Karpinska | Jordan Boyd-Graber | Naoaki Okazaki

pdf bib abs **One Cannot Stand for Everyone! Leveraging Multiple User Simulators to train Task-oriented Dialogue Systems**
Yajiao Liu | Xin Jiang | Yichun Yin | Yasheng Wang | Fei Mi | Qun Liu | Xiang Wan | Benyou Wang

pdf bib abs **SafeConv: Explaining and Correcting Conversational Unsafe Behavior**
Mian Zhang | Lifeng Jin | Linfeng Song | Haitao Mi | Wenliang Chen | Dong Yu

pdf bib abs **Detecting and Mitigating Hallucinations in Machine Translation: Model Internal Workings Alone Do Well, Sentence Similarity Even Better**
David Dale | Elena Voita | Loic Barrault | Marta R. Costa-jussà

pdf bib abs **Explainable Recommendation with Personalized Review Retrieval and Aspect Learning**
Hao Cheng | Shuo Wang | Wensheng Lu | Wei Zhang | Mingyang Zhou | Kezhong Lu | Hao Liao

pdf bib abs **Binary and Ternary Natural Language Generation**
Zechun Liu | Barlas Oguz | Aasish Pappu | Yangyang Shi | Raghuraman Krishnamoorthi

pdf bib abs **Span-Selective Linear Attention Transformers for Effective and Robust Schema-Guided Dialogue State Tracking**
Björn Bebensee | Haejun Lee

pdf bib abs **EM Pre-training for Multi-party Dialogue Response Generation**
Yiyang Li | Hai Zhao

pdf bib abs **ACLM: A Selective-Denoising based Generative Data Augmentation Approach for Low-Resource Complex NER**
Sreyan Ghosh | Utkarsh Tyagi | Manan Suri | Sonal Kumar | Ramaneswaran S | Dinesh Manocha

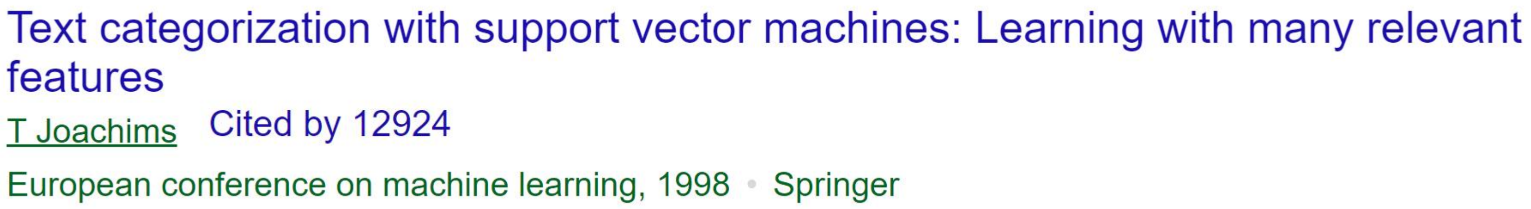
自然语言处理发展：近代史 （1950-2010）

时间	阶段	主要成就
1950年代	早期探索	图灵测试提出
1960-1980年代	规则基础时期	ELIZA等基于规则的对话系统
1990年代	统计方法的革命	隐马尔可夫模型（HMM）等统计模型应用于NLP 支持向量机（SVM）等机器学习算法开始应用
2000年代	机器学习方法的普及	如支持向量机（SVM）、条件随机场（CRF）等在NLP任务中的应用

自然语言处理发展：近代史

时间	阶段	主要成就
1950年代	早期探索	图灵测试提出
1960-1980年代	规则基础时期	ELIZA等基于规则的对话系统
1990年代	统计方法的革命	隐马尔可夫模型（HMM）等统计模型应用于NLP 支持向量机（SVM）等机器学习算法开始应用
2000年代	机器学习方法的普及	如支持向量机（SVM）、条件随机场（CRF）等在NLP任务中的应用

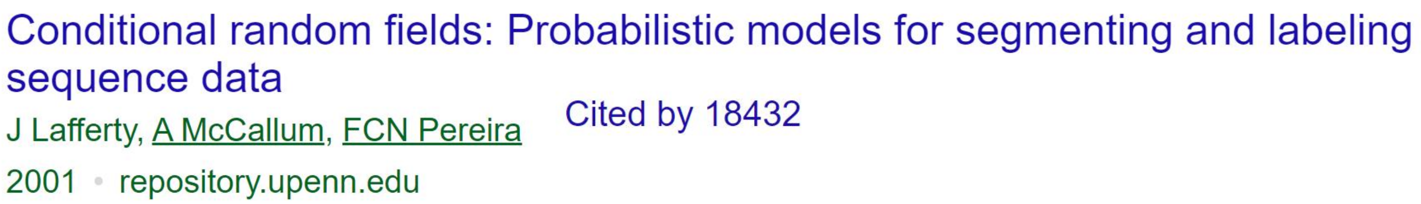
旗舰会议EMNLP会议逐渐形成



Text categorization with support vector machines: Learning with many relevant features

T Joachims Cited by 12924

European conference on machine learning, 1998 • Springer



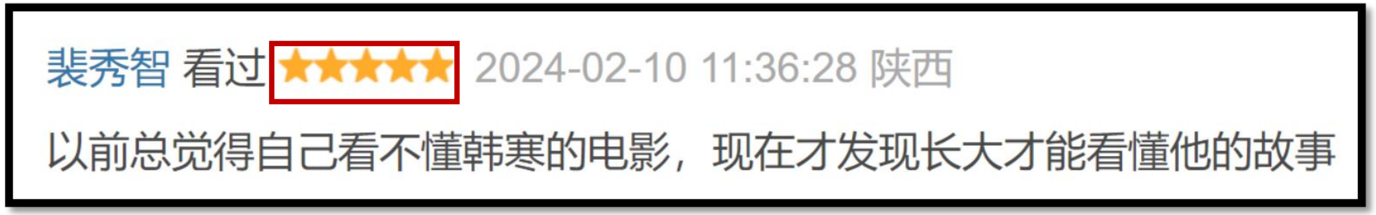
Conditional random fields: Probabilistic models for segmenting and labeling sequence data

J Lafferty, A McCallum, FCN Pereira Cited by 18432

2001 • repository.upenn.edu

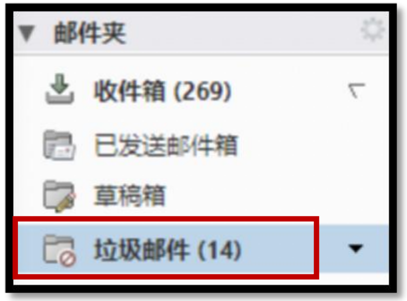
自然语言处理任务：Text to Label

- 任务描述：
  - 输入：一段文本（text）
  - 输出：类别标签（label）
- 具体任务
  - 情感分类
  - 垃圾邮件过滤
- 例子



裴秀智 看过 ★★★★★ 2024-02-10 11:36:28 陕西

以前总觉得自己看不懂韩寒的电影，现在才发现长大才能看懂他的故事



自然语言处理任务：Text-Span to Label

- 任务描述
  - 输入：一个句子(text)和一个词段(span)
  - 输出：类别标签
- 具体任务
  - 基于“视角”的情感分类
- 例子
  - “这款手机的性能超乎我的期待，速度快，**屏幕**😊显示效果也非常好。但是，**电池**😔寿命较短，这让我有些失望。”

自然语言处理任务：Text-Text to Label

□ 任务描述：

- 输入：文本对
- 输出：类别标签

给定两个句子，判断这两者之间的关系是蕴含（entailment）、矛盾（contradiction）还是中立（neutral）

□ 具体任务

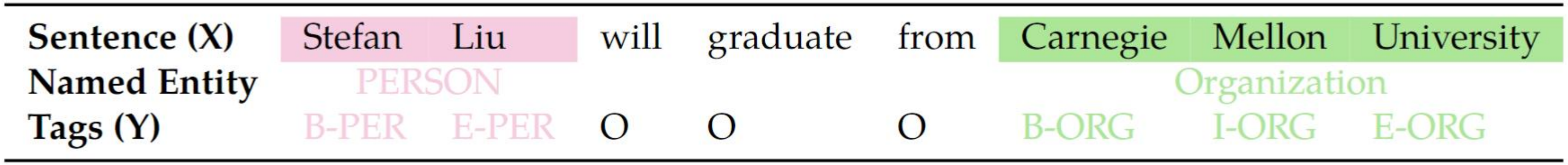
- 自然语言推理 (natural language inference)

□ 例子

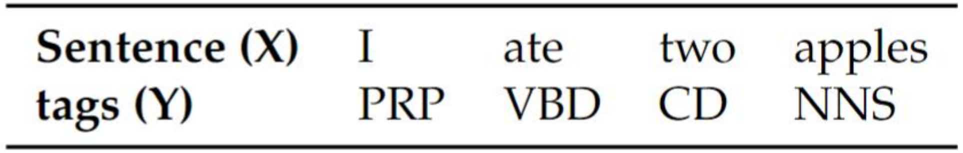
- 句子1：*一群孩子在公园里玩耍。*
- 句子2：
  - □ *孩子们在室内玩耍*（矛盾）
  - □ *孩子们在享受户外活动*（蕴含）
  - □ *公园里正在举行一个生日派对*（中立）

自然语言处理任务：Text to Labels

- 任务描述:
  - 输入：文本
  - 输出：标签序列
- 具体任务
  - 词性标注
  - 命名实体识别
  - 中文分词
- 例子



Sentence (X)	Stefan	Liu	will	graduate	from	Carnegie	Mellon	University
Named Entity	PERSON						Organization	
Tags (Y)	B-PER	E-PER	O	O	O	B-ORG	I-ORG	E-ORG



Sentence (X)	I	ate	two	apples
tags (Y)	PRP	VBD	CD	NNS

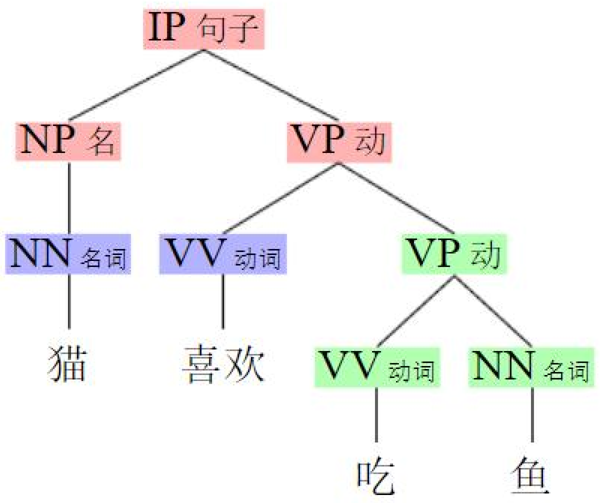
自然语言处理任务：Text to Text

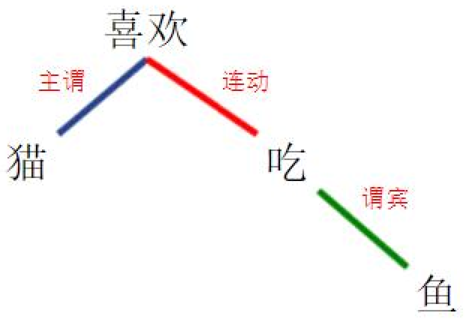
- 任务描述
  - 输入：一段文本
  - 输出：一段文本
- 具体任务
  - 机器翻译
  - 文本摘要
  - 自动对话
- 例子
  - “龙年快乐，万事如意” 的翻译

语言	翻译
英语	Happy Year of the Dragon, may all your wishes come true
西班牙语	Feliz Año del Dragón, que todos tus deseos se hagan realidad
法语	Bonne année du Dragon, que tous vos souhaits se réalisent
日语	龍の年おめでとうございます、万事如意
德语	Glückliches Drachenjahr, mögen alle deine Wünsche in Erfüllung gehen

自然语言处理任务：Text to Tree

- □ 任务描述
  - ■ 输入：一段文本
  - ■ 输出：树结构的标签
- □ 具体任务
  - ■ 短语树：描述的是短语的结构功能
  - ■ 依存树：表示了句子中单词和单词之间的依存关系
- □ 例子

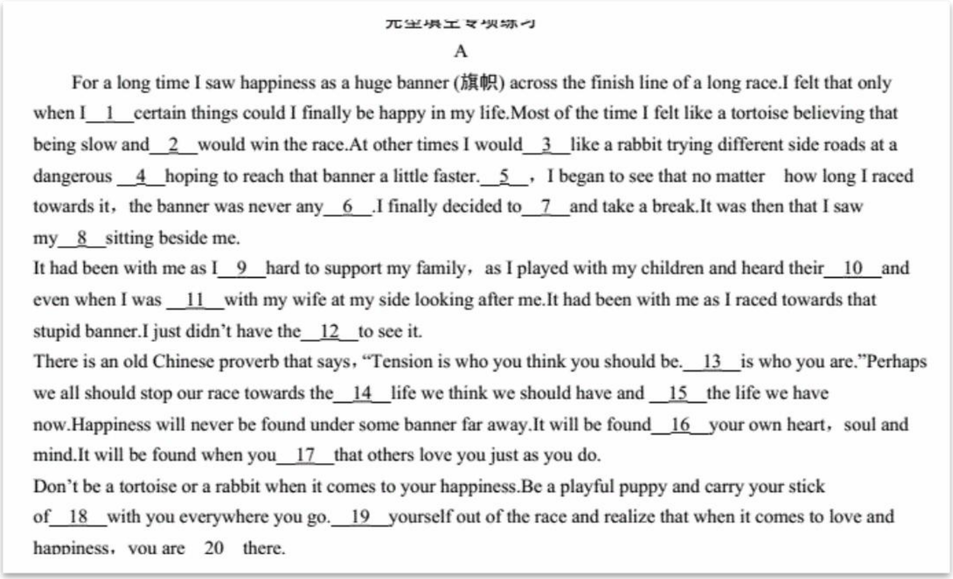




https://opensource.niutrans.com/mtbook/section3-4.html

自然语言处理任务：Word Prediction

- 任务描述：给定一个不完整序列，预测缺失的词元（token）
- 具体任务：
  - 语言模型
  - 词向量学习
- 例子
  - 新年快__



完型填空专项练习

A

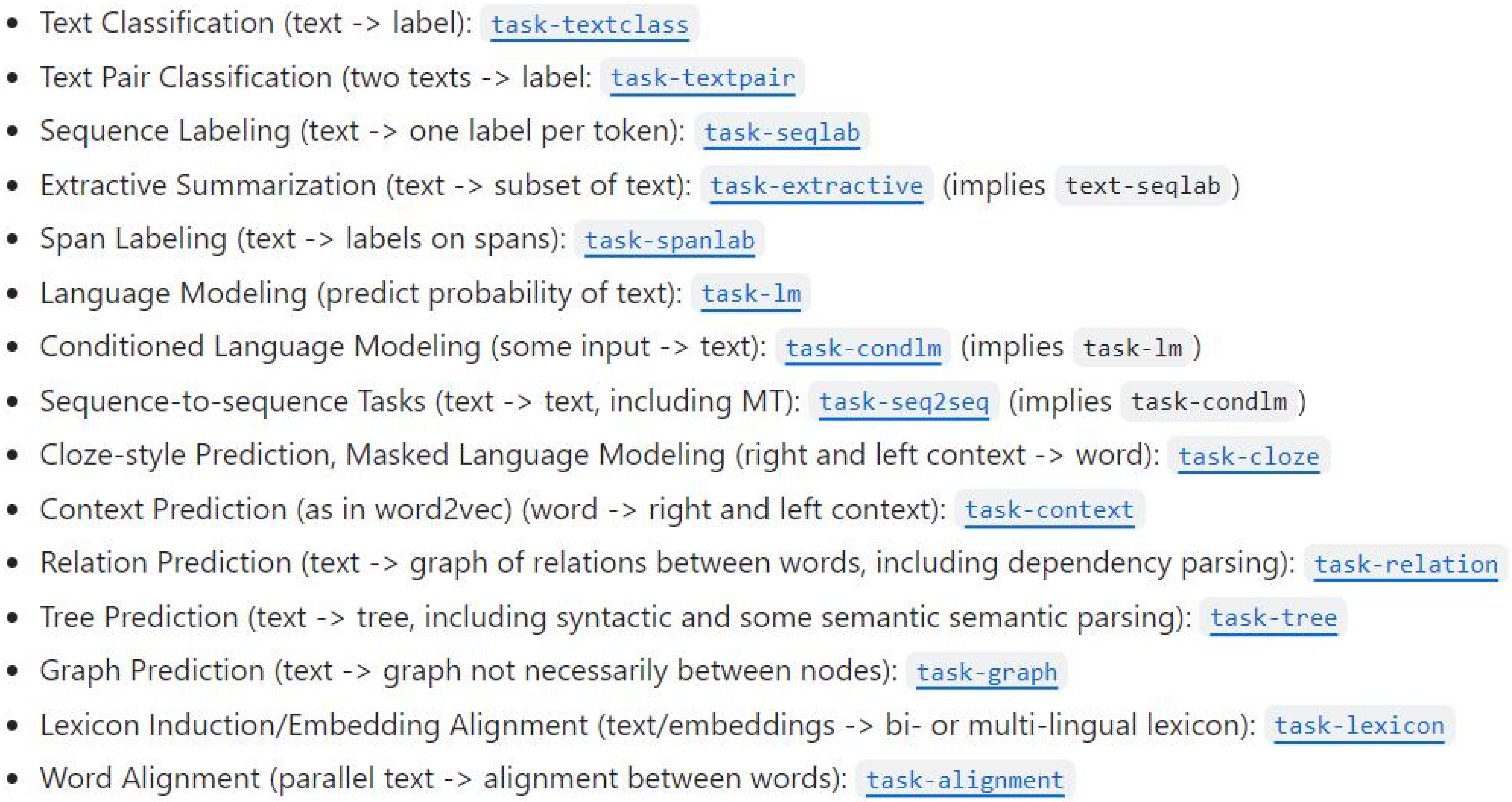
For a long time I saw happiness as a huge banner (旗帜) across the finish line of a long race.I felt that only when I__1__certain things could I finally be happy in my life.Most of the time I felt like a tortoise believing that being slow and__2__would win the race.At other times I would__3__like a rabbit trying different side roads at a dangerous __4__hoping to reach that banner a little faster.__5__， I began to see that no matter how long I raced towards it， the banner was never any__6__.I finally decided to__7__and take a break.It was then that I saw my__8__sitting beside me.

It had been with me as I__9__hard to support my family， as I played with my children and heard their__10__and even when I was __11__with my wife at my side looking after me.It had been with me as I raced towards that stupid banner.I just didn’t have the__12__to see it.

There is an old Chinese proverb that says，“Tension is who you think you should be.__13__is who you are.”Perhaps we all should stop our race towards the__14__life we think we should have and __15__the life we have now.Happiness will never be found under some banner far away.It will be found__16__your own heart， soul and mind.It will be found when you__17__that others love you just as you do.

Don’t be a tortoise or a rabbit when it comes to your happiness.Be a playful puppy and carry your stick of__18__with you everywhere you go.__19__yourself out of the race and realize that when it comes to love and happiness， you are 20 there.

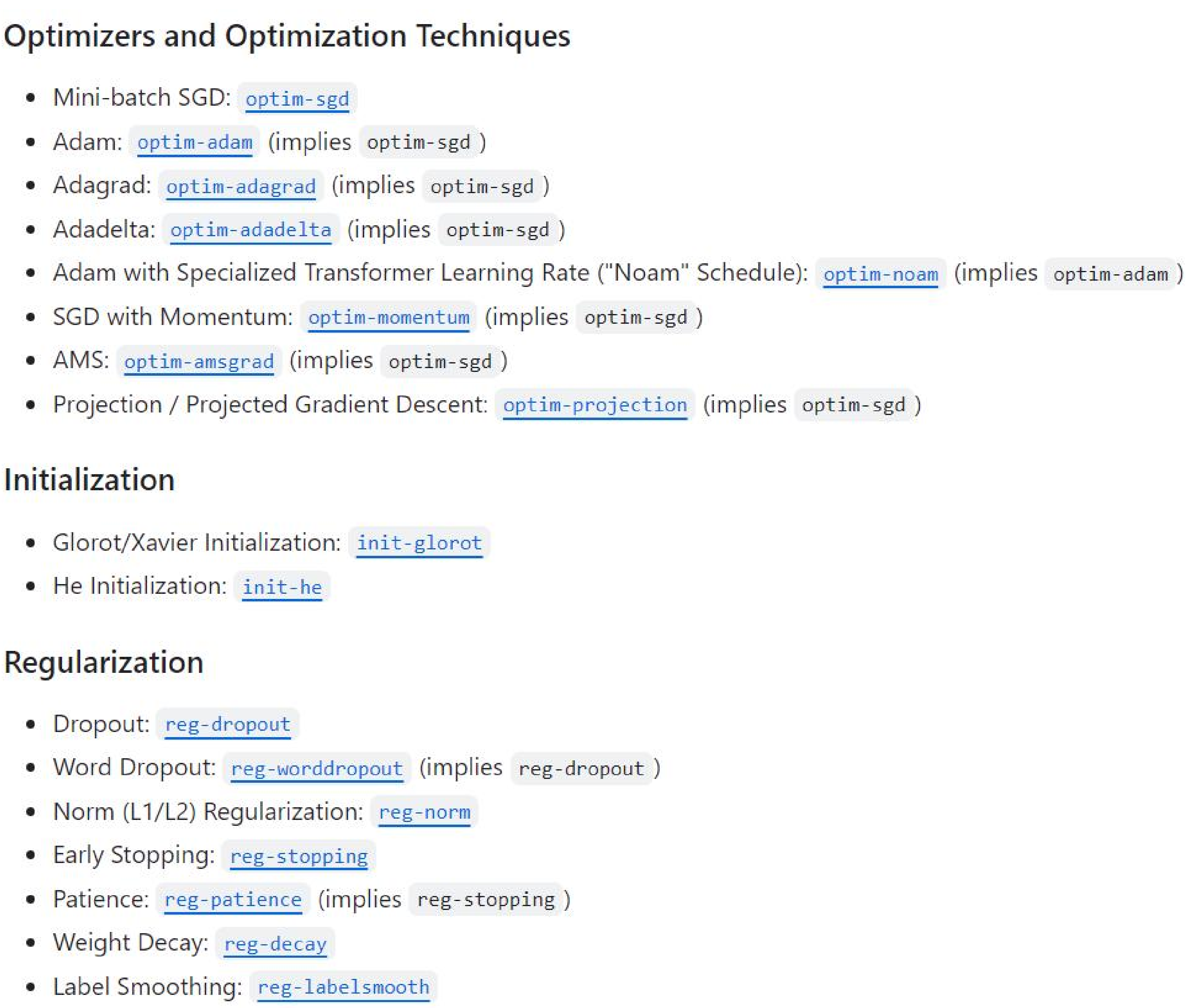
自然语言处理中一些重要的概念：Prediction Task



- Text Classification (text -> label): `task-textclass`
- Text Pair Classification (two texts -> label: `task-textpair`
- Sequence Labeling (text -> one label per token): `task-seqlab`
- Extractive Summarization (text -> subset of text): `task-extractive` (implies `text-seqlab`)
- Span Labeling (text -> labels on spans): `task-spanlab`
- Language Modeling (predict probability of text): `task-lm`
- Conditioned Language Modeling (some input -> text): `task-condlm` (implies `task-lm`)
- Sequence-to-sequence Tasks (text -> text, including MT): `task-seq2seq` (implies `task-condlm`)
- Cloze-style Prediction, Masked Language Modeling (right and left context -> word): `task-cloze`
- Context Prediction (as in word2vec) (word -> right and left context): `task-context`
- Relation Prediction (text -> graph of relations between words, including dependency parsing): `task-relation`
- Tree Prediction (text -> tree, including syntactic and some semantic semantic parsing): `task-tree`
- Graph Prediction (text -> graph not necessarily between nodes): `task-graph`
- Lexicon Induction/Embedding Alignment (text/embeddings -> bi- or multi-lingual lexicon): `task-lexicon`
- Word Alignment (parallel text -> alignment between words): `task-alignment`

https://github.com/neulab/nn4nlp-concepts/blob/master/concepts.md

自然语言处理中一些重要的概念：Optimization/Learning



Optimizers and Optimization Techniques

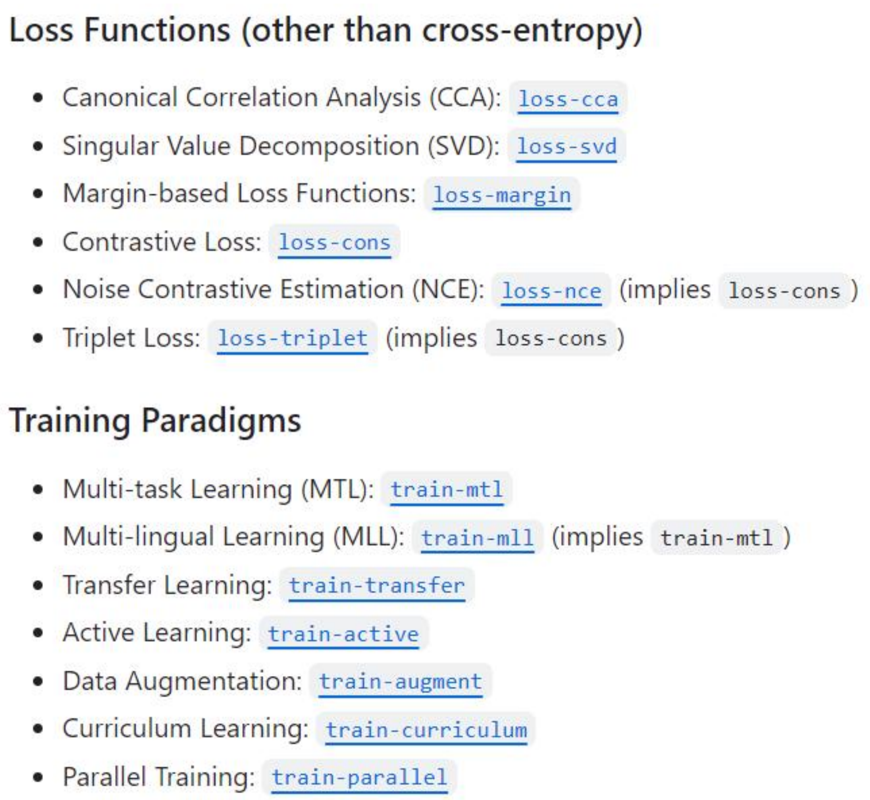
- Mini-batch SGD: `optim-sgd`
- Adam: `optim-adam` (implies `optim-sgd`)
- Adagrad: `optim-adagrad` (implies `optim-sgd`)
- Adadelta: `optim-adadelta` (implies `optim-sgd`)
- Adam with Specialized Transformer Learning Rate ("Noam" Schedule): `optim-noam` (implies `optim-adam`)
- SGD with Momentum: `optim-momentum` (implies `optim-sgd`)
- AMS: `optim-amsgrad` (implies `optim-sgd`)
- Projection / Projected Gradient Descent: `optim-projection` (implies `optim-sgd`)

Initialization

- Glorot/Xavier Initialization: `init-glorot`
- He Initialization: `init-he`

Regularization

- Dropout: `reg-dropout`
- Word Dropout: `reg-worddropout` (implies `reg-dropout`)
- Norm (L1/L2) Regularization: `reg-norm`
- Early Stopping: `reg-stopping`
- Patience: `reg-patience` (implies `reg-stopping`)
- Weight Decay: `reg-decay`
- Label Smoothing: `reg-labelsmooth`



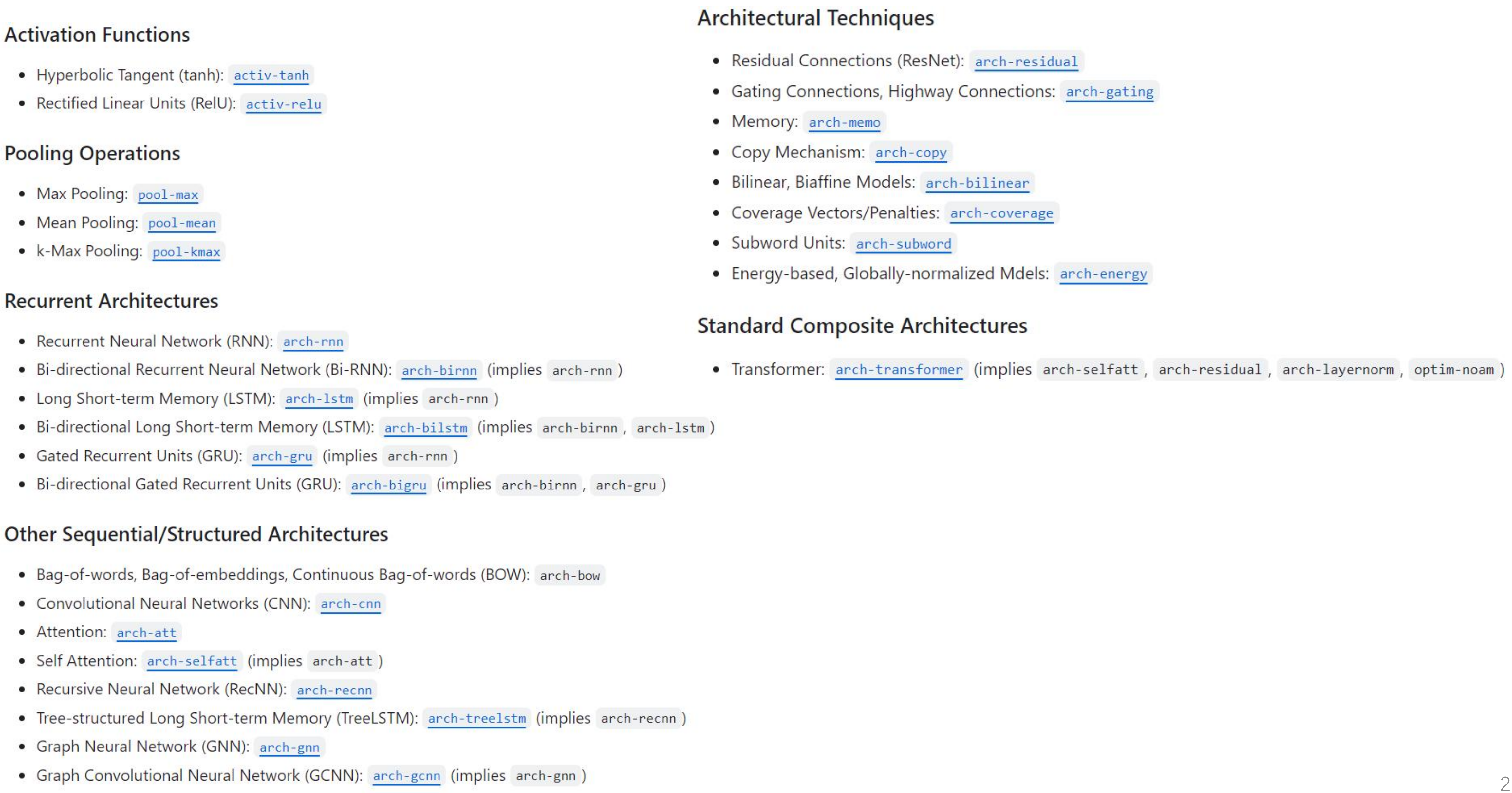
Loss Functions (other than cross-entropy)

- Canonical Correlation Analysis (CCA): `loss-cca`
- Singular Value Decomposition (SVD): `loss-svd`
- Margin-based Loss Functions: `loss-margin`
- Contrastive Loss: `loss-cons`
- Noise Contrastive Estimation (NCE): `loss-nce` (implies `loss-cons`)
- Triplet Loss: `loss-triplet` (implies `loss-cons`)

Training Paradigms

- Multi-task Learning (MTL): `train-mtl`
- Multi-lingual Learning (MLL): `train-mll` (implies `train-mtl`)
- Transfer Learning: `train-transfer`
- Active Learning: `train-active`
- Data Augmentation: `train-augment`
- Curriculum Learning: `train-curriculum`
- Parallel Training: `train-parallel`

自然语言处理中一些重要的概念: Neural Arcs



Activation Functions

- Hyperbolic Tangent (tanh): `activ-tanh`
- Rectified Linear Units (RelU): `activ-relu`

Pooling Operations

- Max Pooling: `pool-max`
- Mean Pooling: `pool-mean`
- k-Max Pooling: `pool-kmax`

Recurrent Architectures

- Recurrent Neural Network (RNN): `arch-rnn`
- Bi-directional Recurrent Neural Network (Bi-RNN): `arch-birnn` (implies `arch-rnn`)
- Long Short-term Memory (LSTM): `arch-lstm` (implies `arch-rnn`)
- Bi-directional Long Short-term Memory (LSTM): `arch-bilstm` (implies `arch-birnn` , `arch-lstm`)
- Gated Recurrent Units (GRU): `arch-gru` (implies `arch-rnn`)
- Bi-directional Gated Recurrent Units (GRU): `arch-bigru` (implies `arch-birnn` , `arch-gru`)

Other Sequential/Structured Architectures

- Bag-of-words, Bag-of-embeddings, Continuous Bag-of-words (BOW): `arch-bow`
- Convolutional Neural Networks (CNN): `arch-cnn`
- Attention: `arch-att`
- Self Attention: `arch-selfatt` (implies `arch-att`)
- Recursive Neural Network (RecNN): `arch-recnn`
- Tree-structured Long Short-term Memory (TreeLSTM): `arch-treelstm` (implies `arch-recnn`)
- Graph Neural Network (GNN): `arch-gnn`
- Graph Convolutional Neural Network (GCNN): `arch-gcnn` (implies `arch-gnn`)

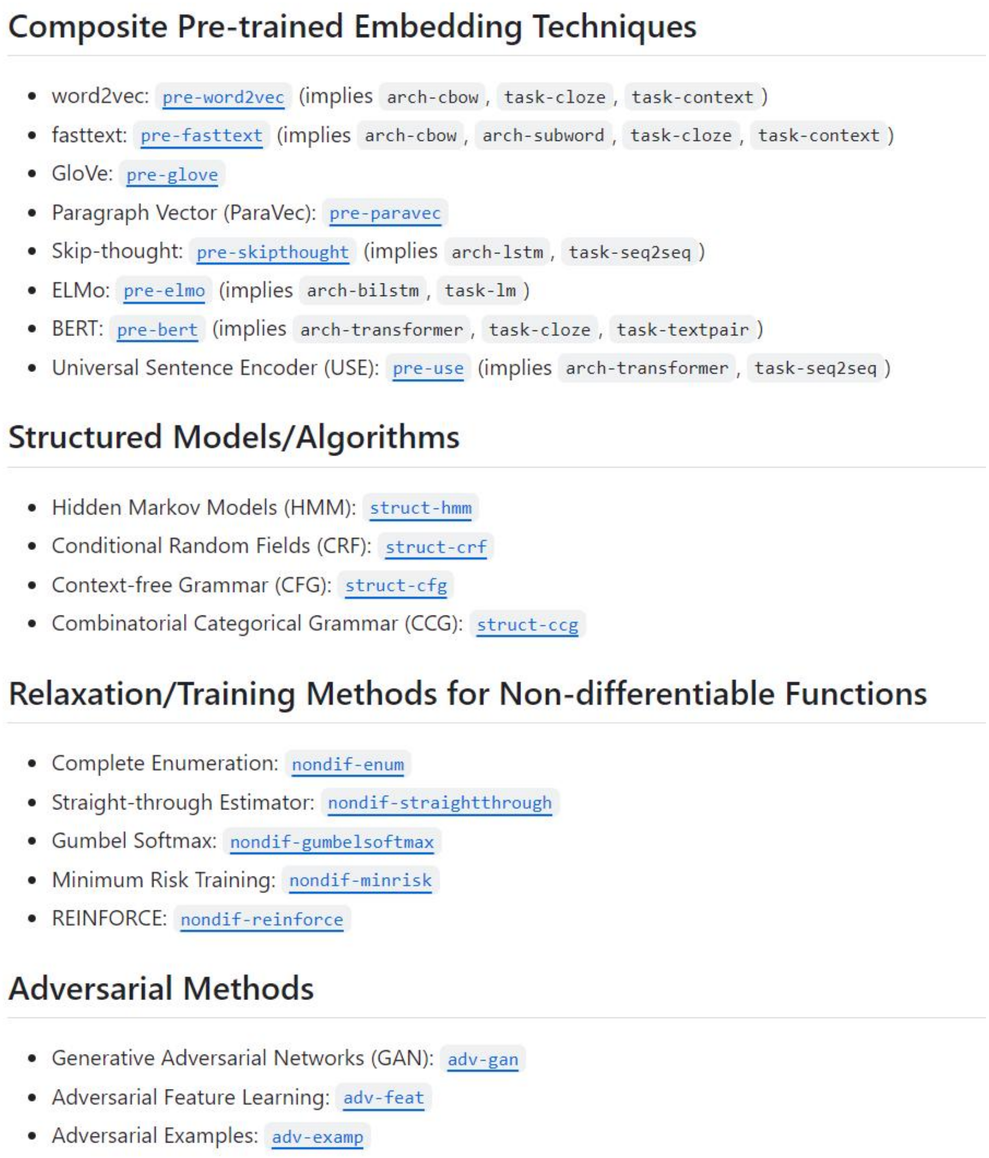
Architectural Techniques

- Residual Connections (ResNet): `arch-residual`
- Gating Connections, Highway Connections: `arch-gating`
- Memory: `arch-memo`
- Copy Mechanism: `arch-copy`
- Bilinear, Biaffine Models: `arch-bilinear`
- Coverage Vectors/Penalties: `arch-coverage`
- Subword Units: `arch-subword`
- Energy-based, Globally-normalized Mdels: `arch-energy`

Standard Composite Architectures

- Transformer: `arch-transformer` (implies `arch-selfatt` , `arch-residual` , `arch-layernorm` , `optim-noam`)

自然语言处理中一些重要的概念: 其它



Composite Pre-trained Embedding Techniques

- word2vec: `pre-word2vec` (implies `arch-cbow`, `task-cloze`, `task-context`)
- fasttext: `pre-fasttext` (implies `arch-cbow`, `arch-subword`, `task-cloze`, `task-context`)
- GloVe: `pre-glove`
- Paragraph Vector (ParaVec): `pre-paravec`
- Skip-thought: `pre-skipthought` (implies `arch-lstm`, `task-seq2seq`)
- ELMo: `pre-elmo` (implies `arch-bilstm`, `task-lm`)
- BERT: `pre-bert` (implies `arch-transformer`, `task-cloze`, `task-textpair`)
- Universal Sentence Encoder (USE): `pre-use` (implies `arch-transformer`, `task-seq2seq`)

Structured Models/Algorithms

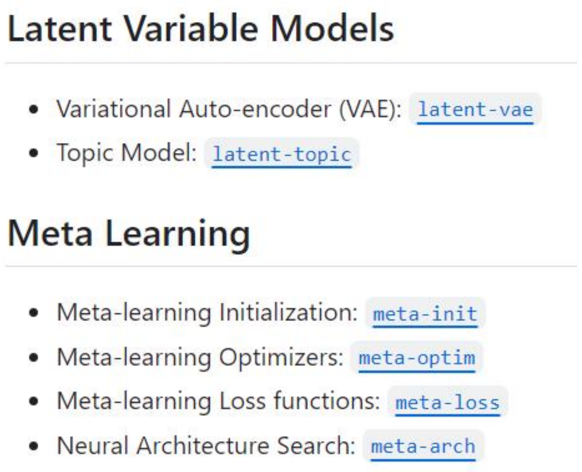
- Hidden Markov Models (HMM): `struct-hmm`
- Conditional Random Fields (CRF): `struct-crf`
- Context-free Grammar (CFG): `struct-cfg`
- Combinatorial Categorical Grammar (CCG): `struct-ccg`

Relaxation/Training Methods for Non-differentiable Functions

- Complete Enumeration: `nondif-enum`
- Straight-through Estimator: `nondif-straightthrough`
- Gumbel Softmax: `nondif-gumbelsoftmax`
- Minimum Risk Training: `nondif-minrisk`
- REINFORCE: `nondif-reinforce`

Adversarial Methods

- Generative Adversarial Networks (GAN): `adv-gan`
- Adversarial Feature Learning: `adv-feat`
- Adversarial Examples: `adv-examp`



Latent Variable Models

- Variational Auto-encoder (VAE): `latent-vae`
- Topic Model: `latent-topic`

Meta Learning

- Meta-learning Initialization: `meta-init`
- Meta-learning Optimizers: `meta-optim`
- Meta-learning Loss functions: `meta-loss`
- Neural Architecture Search: `meta-arch`

自然语言处理发展：现代史 （2010-现在）

- □ **Feature Engineering**
- □ Architecture Engineering
- □ Objective Engineering
- □ Prompt Engineering

- **Paradigm**： Fully Supervised Learning (Non-neural Network)
- **Date**： Before 2013
- **Characteristic**： Traditional machine learning model is mainly used, which requires manual feature definition of input text
- **Typical Work**：
  - CRF (Conditional Random Field)

自然语言处理发展：现代史 （2010-现在）

- □ Feature Engineering
- □ **Architecture Engineering**
- □ Objective Engineering
- □ Prompt Engineering

- **Paradigm**： Fully Supervised Learning (Neural Network)
- **Date**： 2013 - 2018
- **Characteristic**：
  - Rely on neural networks
  - Do not need to manually define features, but should explore the network structure （e.g.：LSTM v.s CNN）
- **Typical Work**：
  - CNN for Text Classification

自然语言处理发展：现代史 （2010-现在）

- □ Feature Engineering
- □ Architecture Engineering
- □ **Objective Engineering**
- □ Prompt Engineering

- **Paradigm**： Pre-train, Fine-tune
- **Date**： 2018-Now
- **Characteristic**：
  - context-dependent PLMs
  - Need to pay attention to the definition and selection of objective functions
- **Typical Work**： BERT

自然语言处理发展：现代史 （2010-现在）

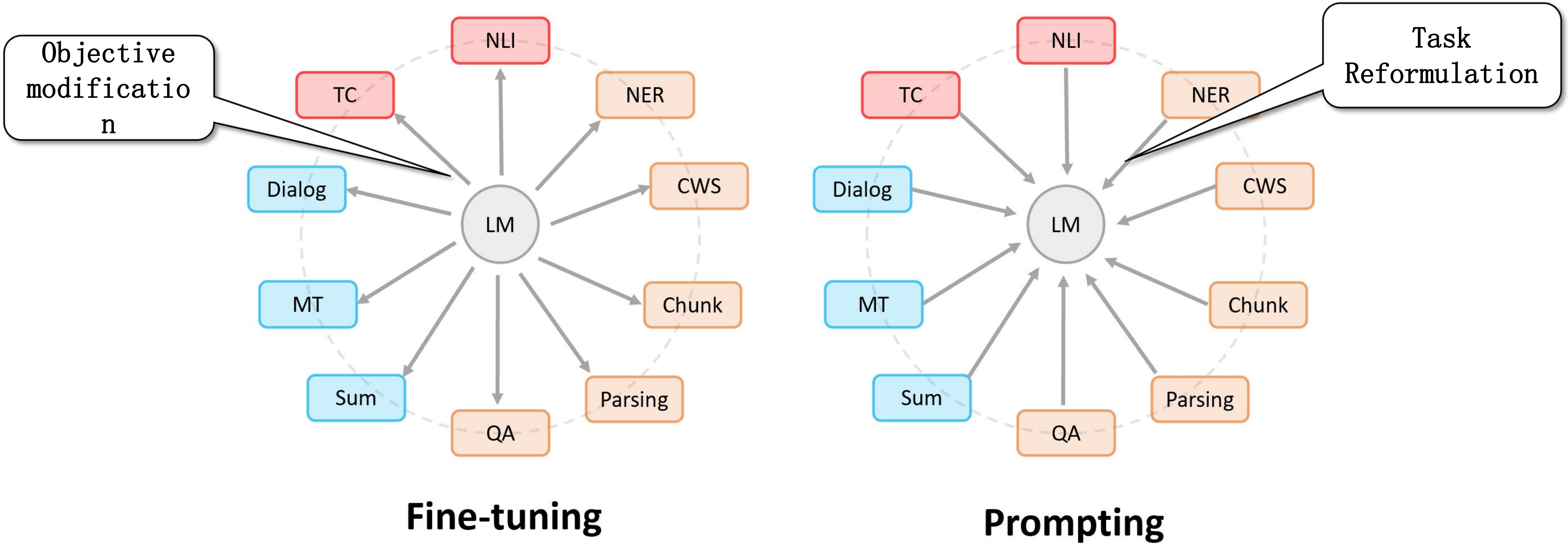
- □ Feature Engineering
- □ Architecture Engineering
- □ Objective Engineering
- □ **Prompt Engineering**

- **Paradigm**： Pre-train, Prompt, Predict
- **Date**： 2019-Now
- **Characteristic**：
  - NLP tasks are modeled entirely by relying on PLMs
  - More efforts on prompt design
- **Typical Work:** GPT3

NLP技术重要概念的变迁史

https://public.flourish.studio/visualisation/16875387/

任务的“大一统”

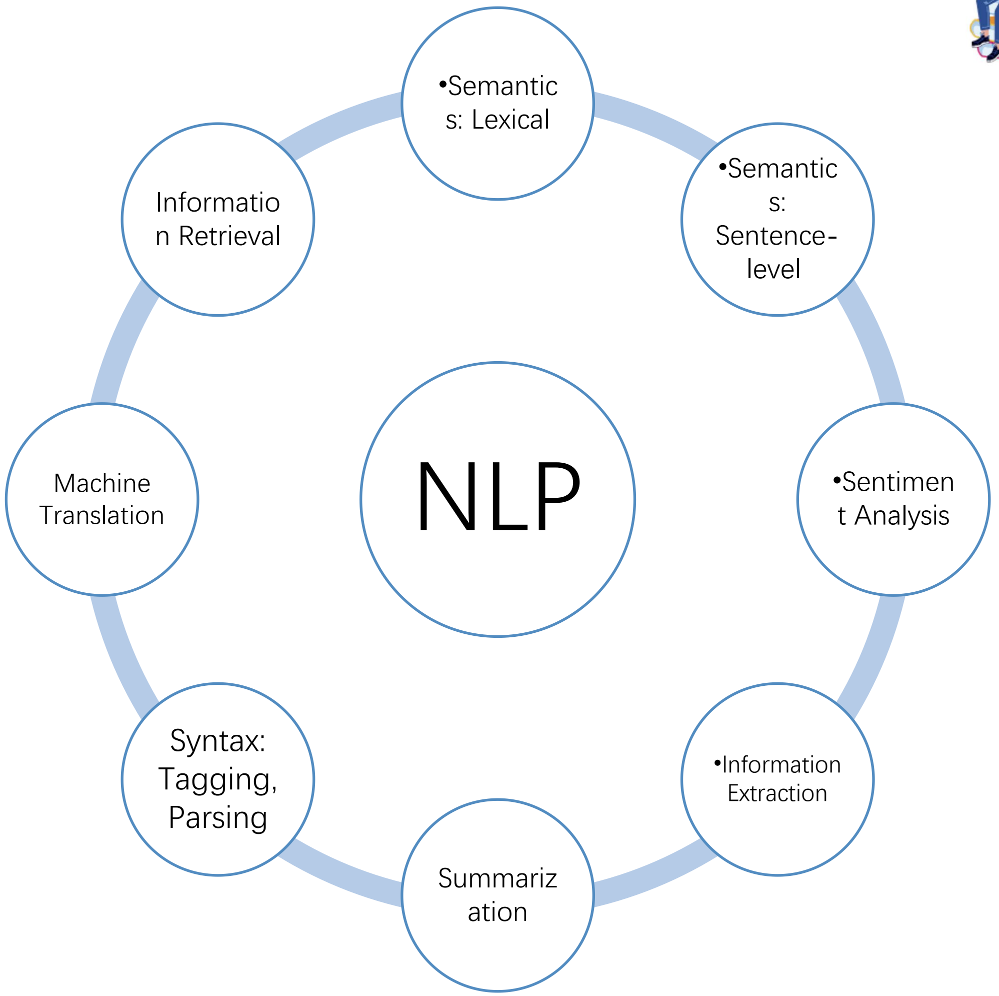


Fine-tuning

Prompting

技术范式变革 推动 科研范式改变

过去：任务划分



NLP

•Semantics: Lexical

•Semantics: Sentence-level

•Sentiment Analysis

•Information Extraction

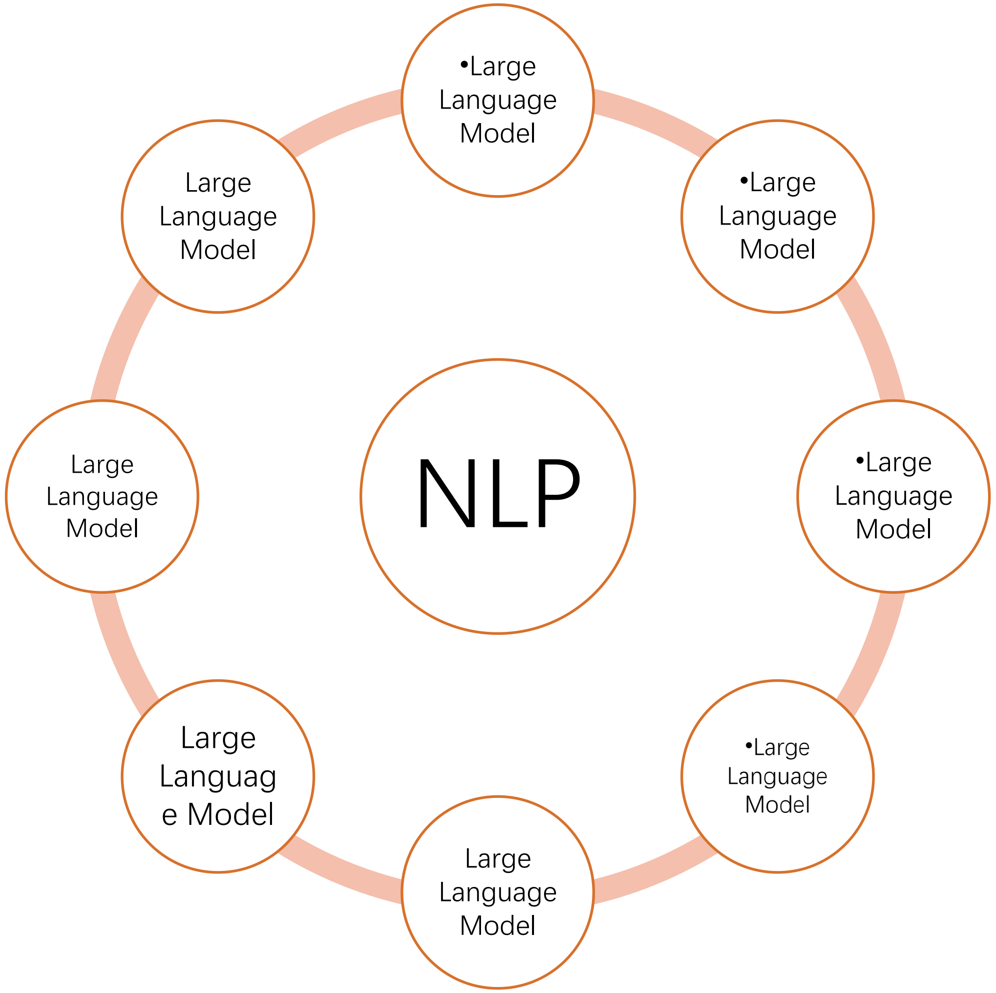
Summarization

Syntax: Tagging, Parsing

Machine Translation

Information Retrieval

现在：走向大模型



NLP

•Large Language Model

•Large Language Model

•Large Language Model

•Large Language Model

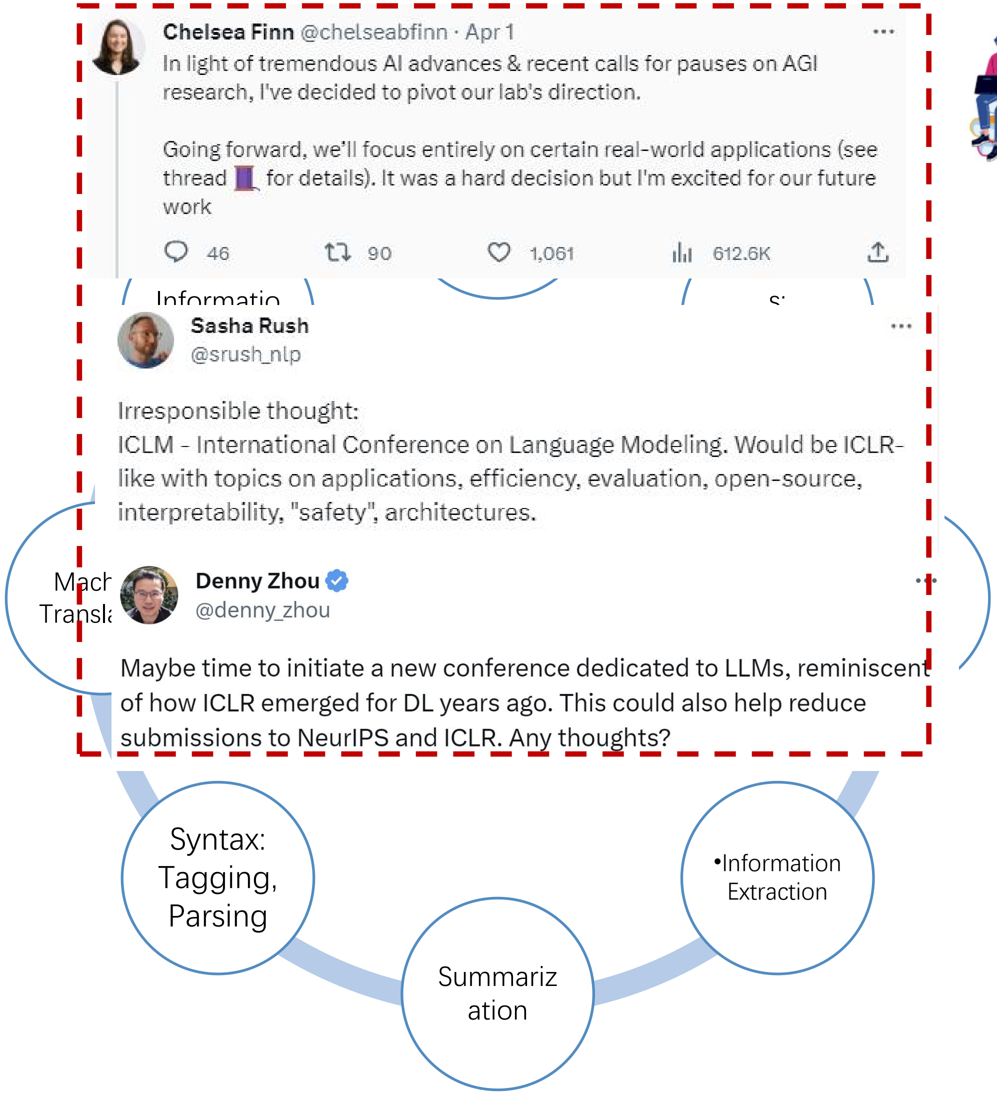
Large Language Model

Large Language Model

Large Language Model

Large Language Model

技术范式变革 推动 科研范式改变



Chelsea Finn @chelseabfinn · Apr 1

In light of tremendous AI advances & recent calls for pauses on AGI research, I've decided to pivot our lab's direction.

Going forward, we'll focus entirely on certain real-world applications (see thread 🧵 for details). It was a hard decision but I'm excited for our future work

46 90 1,061 612.6K

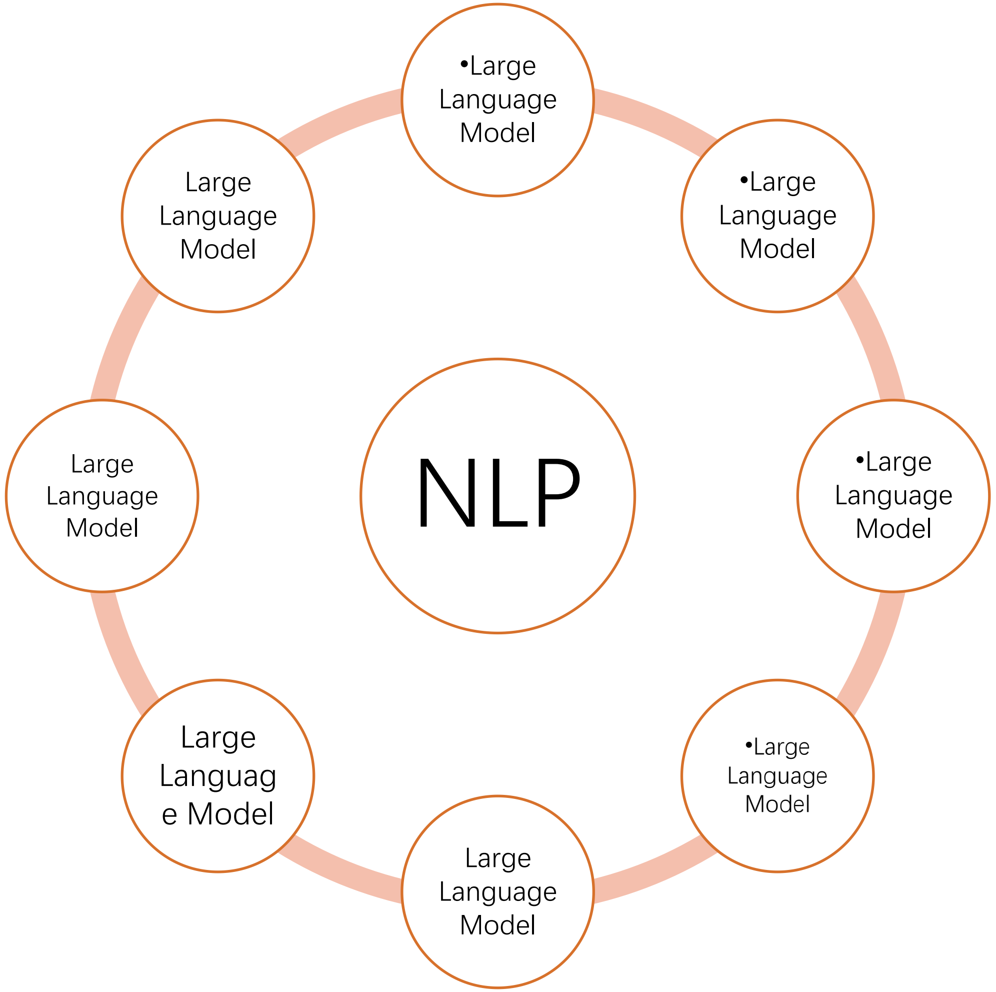
Sasha Rush @srush_nlp

Irresponsible thought:
ICLM - International Conference on Language Modeling. Would be ICLR-like with topics on applications, efficiency, evaluation, open-source, interpretability, "safety", architectures.

Denny Zhou @denny_zhou

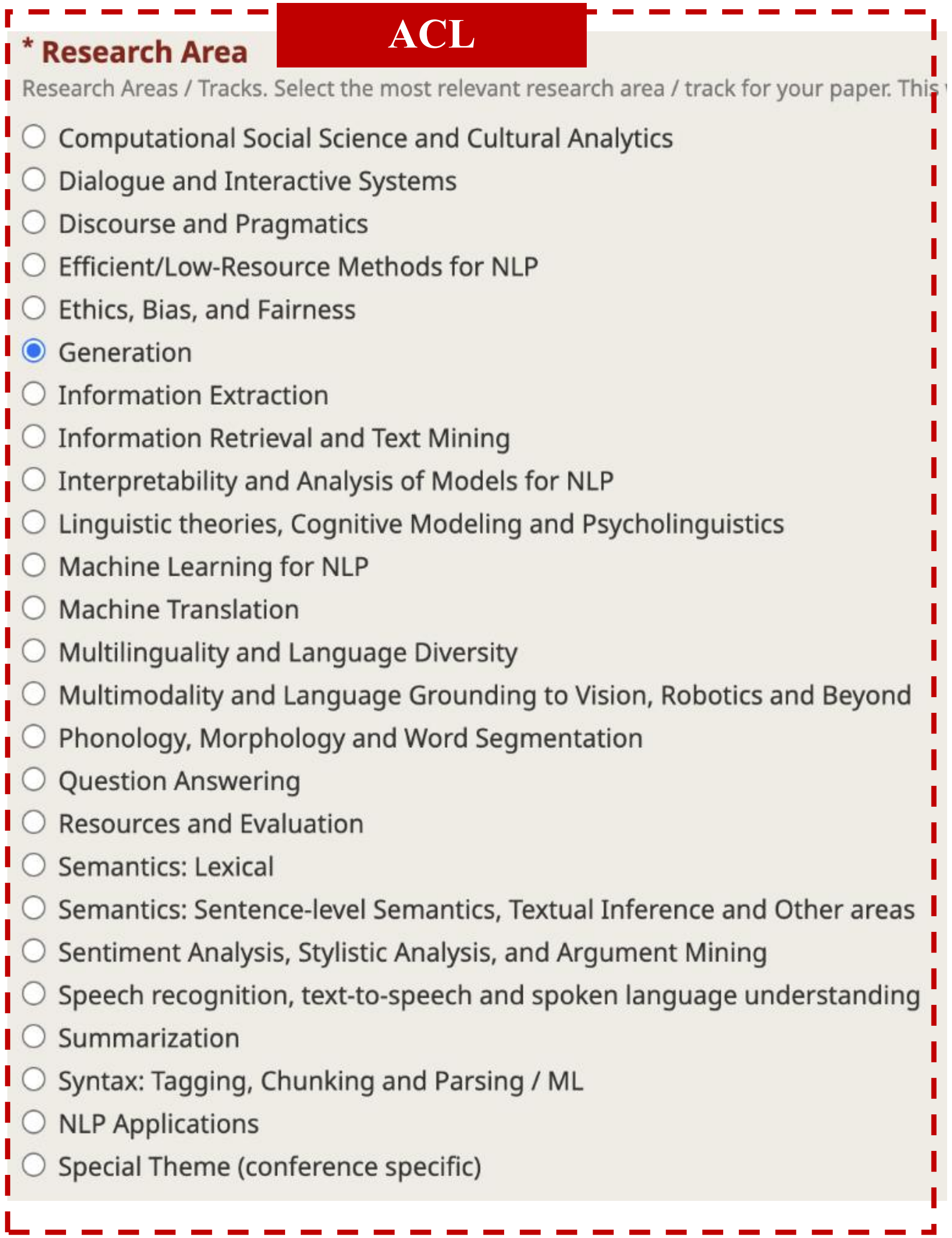
Maybe time to initiate a new conference dedicated to LLMs, reminiscent of how ICLR emerged for DL years ago. This could also help reduce submissions to NeurIPS and ICLR. Any thoughts?

现在：走向大模型



技术范式变革 推动 科研范式改变

ACL

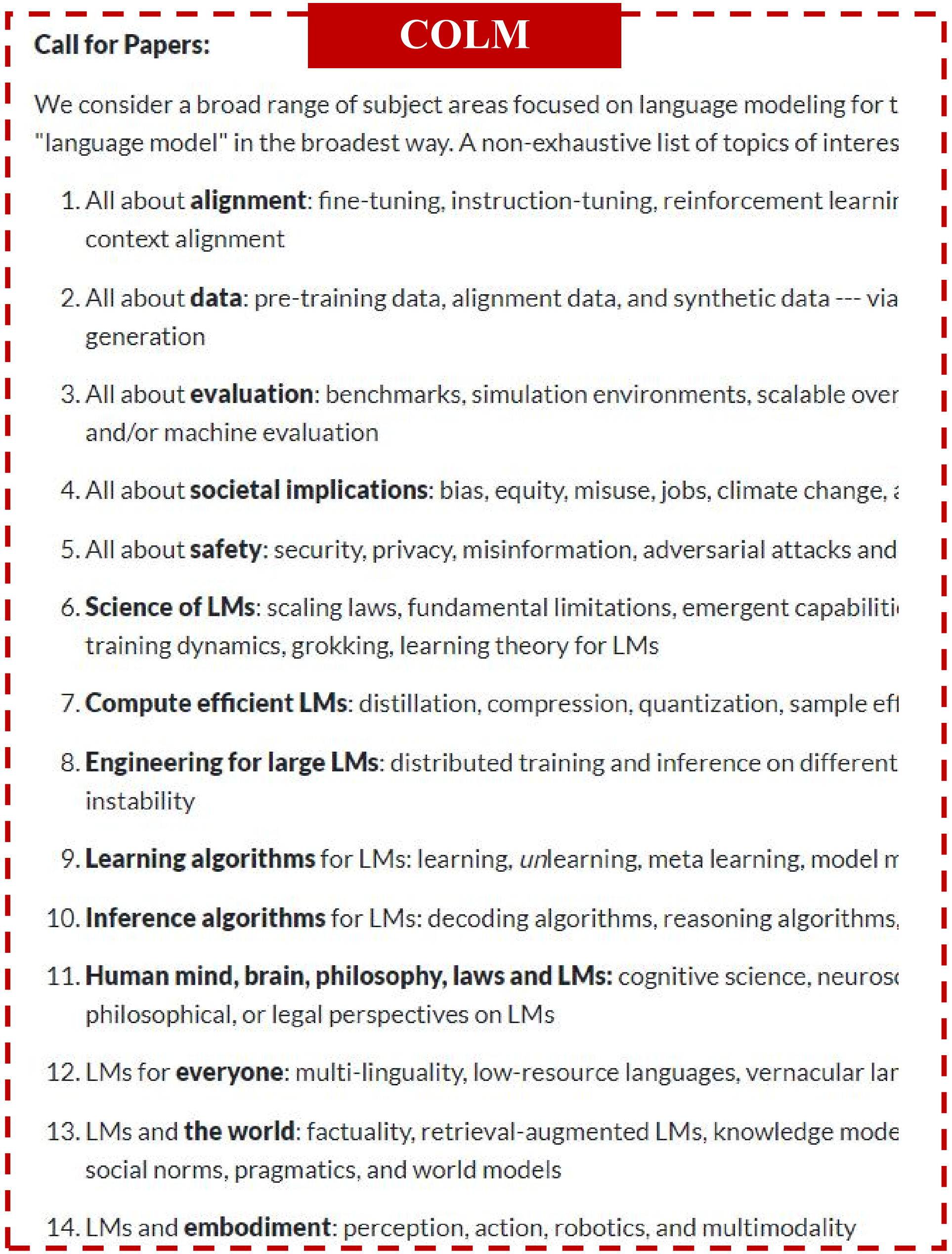


* Research Area

Research Areas / Tracks. Select the most relevant research area / track for your paper. This

○ Computational Social Science and Cultural Analytics
○ Dialogue and Interactive Systems
○ Discourse and Pragmatics
○ Efficient/Low-Resource Methods for NLP
○ Ethics, Bias, and Fairness
● Generation
○ Information Extraction
○ Information Retrieval and Text Mining
○ Interpretability and Analysis of Models for NLP
○ Linguistic theories, Cognitive Modeling and Psycholinguistics
○ Machine Learning for NLP
○ Machine Translation
○ Multilinguality and Language Diversity
○ Multimodality and Language Grounding to Vision, Robotics and Beyond
○ Phonology, Morphology and Word Segmentation
○ Question Answering
○ Resources and Evaluation
○ Semantics: Lexical
○ Semantics: Sentence-level Semantics, Textual Inference and Other areas
○ Sentiment Analysis, Stylistic Analysis, and Argument Mining
○ Speech recognition, text-to-speech and spoken language understanding
○ Summarization
○ Syntax: Tagging, Chunking and Parsing / ML
○ NLP Applications
○ Special Theme (conference specific)

COLM



Call for Papers:

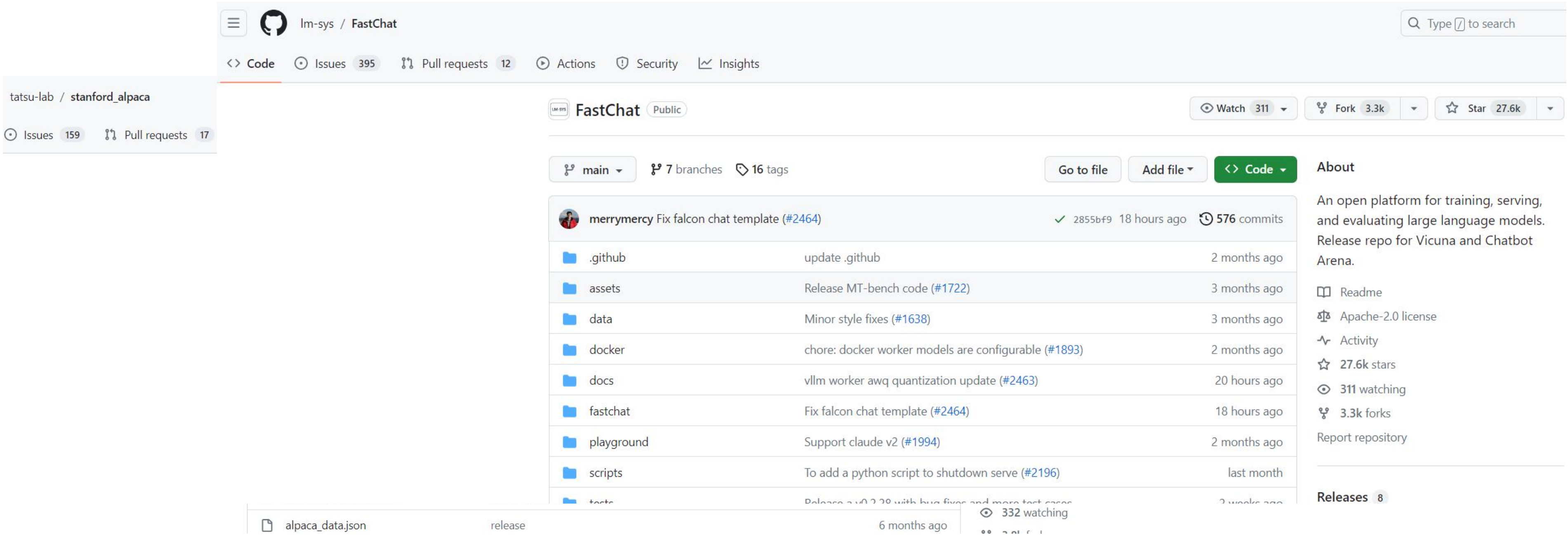
We consider a broad range of subject areas focused on language modeling for t
"language model" in the broadest way. A non-exhaustive list of topics of interes

1. All about **alignment**: fine-tuning, instruction-tuning, reinforcement learnir context alignment
2. All about **data**: pre-training data, alignment data, and synthetic data --- via generation
3. All about **evaluation**: benchmarks, simulation environments, scalable over and/or machine evaluation
4. All about **societal implications**: bias, equity, misuse, jobs, climate change, a
5. All about **safety**: security, privacy, misinformation, adversarial attacks and
6. **Science of LMs**: scaling laws, fundamental limitations, emergent capabiliti training dynamics, grokking, learning theory for LMs
7. **Compute efficient LMs**: distillation, compression, quantization, sample eff
8. **Engineering for large LMs**: distributed training and inference on different instability
9. **Learning algorithms** for LMs: learning, *un*learning, meta learning, model m
10. **Inference algorithms** for LMs: decoding algorithms, reasoning algorithms,
11. **Human mind, brain, philosophy, laws and LMs:** cognitive science, neurosc philosophical, or legal perspectives on LMs
12. LMs for **everyone**: multi-linguality, low-resource languages, vernacular lar
13. LMs and **the world**: factuality, retrieval-augmented LMs, knowledge mode social norms, pragmatics, and world models
14. LMs and **embodiment**: perception, action, robotics, and multimodality

技术范式变革 推动 科研范式改变

最新技术获取源发生变化

- 10年前: ACL/EMNLP/ICML
- 5年前：ArXiv
- 现在：Github/Twitter/最新博文



merrymercy Fix falcon chat template (#2464) ✓ 2855bf9 18 hours ago 576 commits

.github	update .github	2 months ago
assets	Release MT-bench code (#1722)	3 months ago
data	Minor style fixes (#1638)	3 months ago
docker	chore: docker worker models are configurable (#1893)	2 months ago
docs	vllm worker awq quantization update (#2463)	20 hours ago
fastchat	Fix falcon chat template (#2464)	18 hours ago
playground	Support claude v2 (#1994)	2 months ago
scripts	To add a python script to shutdown serve (#2196)	last month
tests	Release a v0.2.28 with bug fixes and more test cases	2 weeks ago
alpaca_data.json	release	6 months ago

About

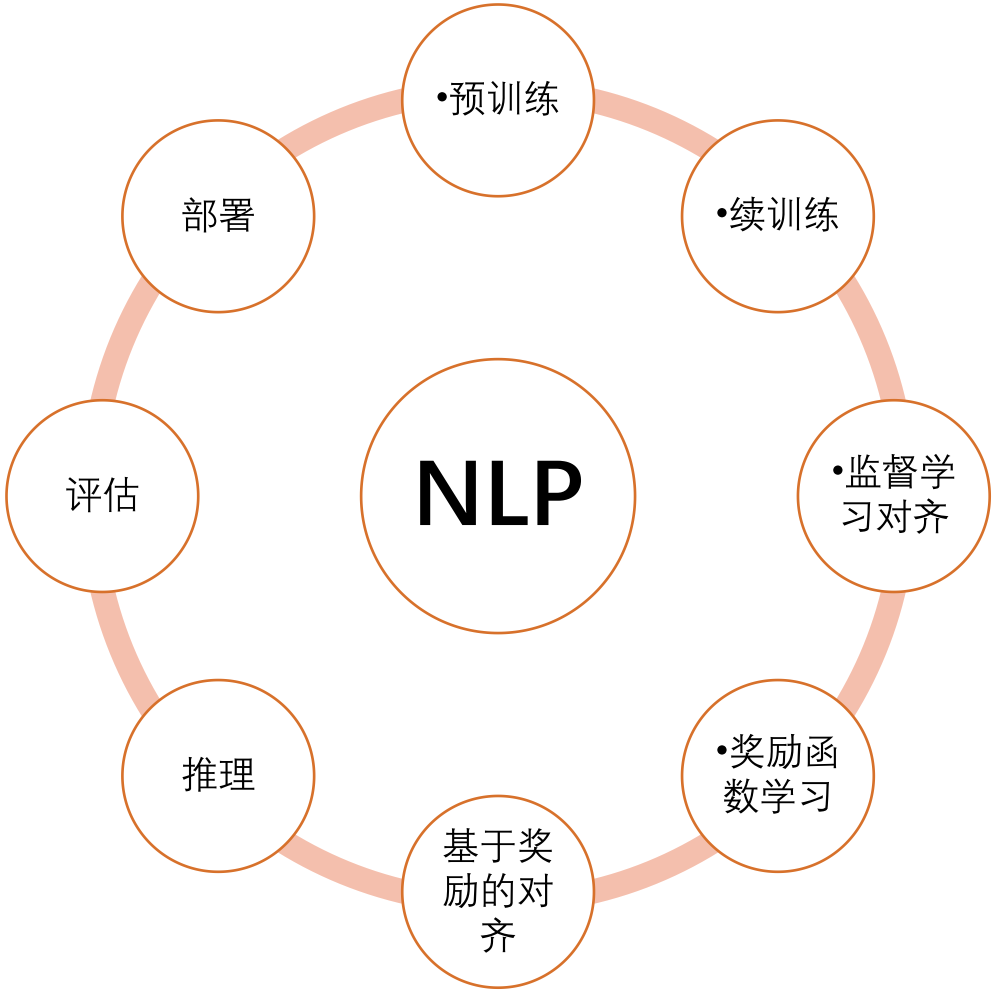
An open platform for training, serving, and evaluating large language models. Release repo for Vicuna and Chatbot Arena.

Readme
Apache-2.0 license
Activity
27.6k stars
311 watching
3.3k forks
Report repository

Releases 8

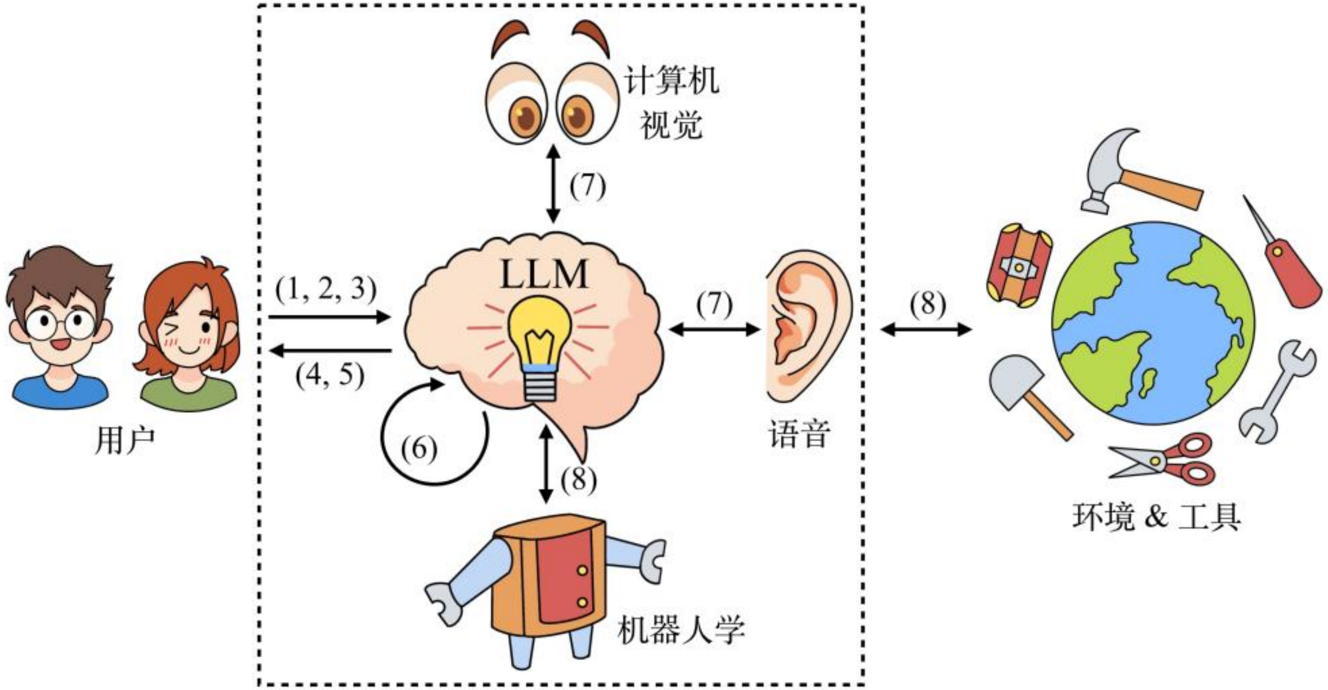
332 watching

大模型在做什么?



技术栈视角（LLMOps）

大模型在做什么?

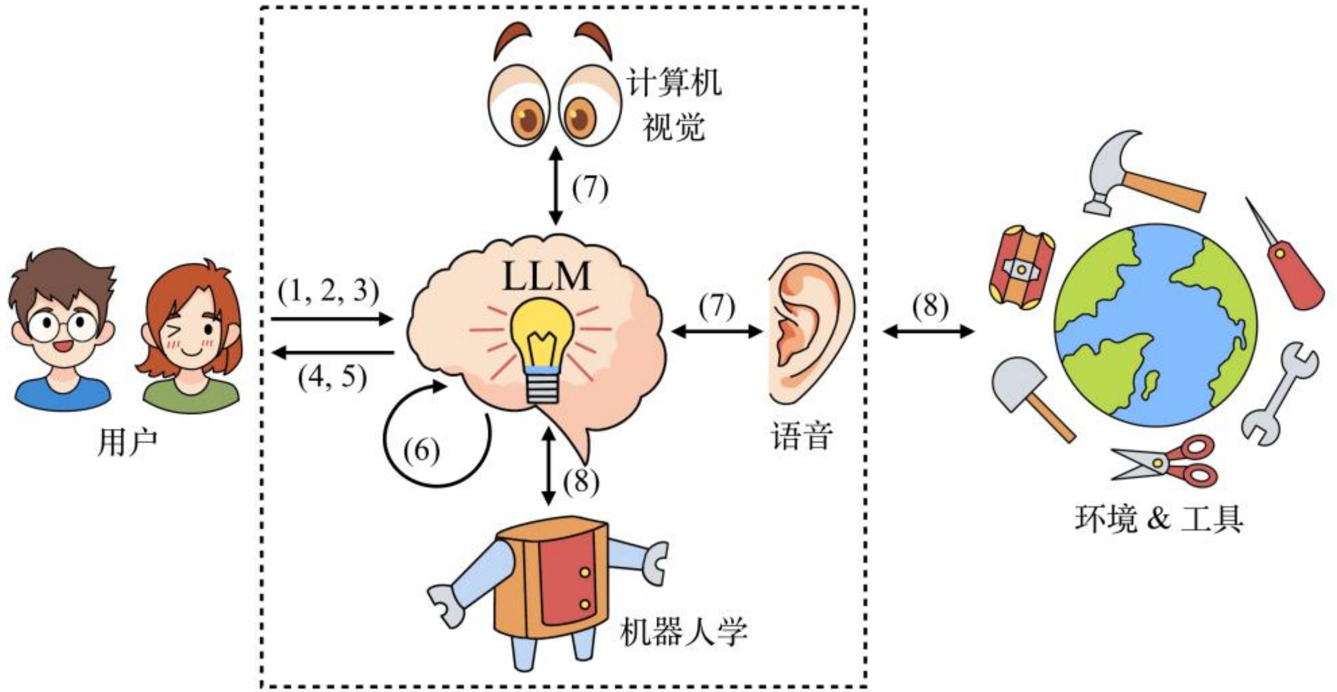


全景技术栈

生成式AI

1. 需求挖掘
2. 提示工程
3. 系统安全
4. 用户接口
5. AI安全 & 对齐
6. LLMOps
7. 多模态功能
8. 环境交互性

新的时代如何展开科研?



生成式AI

1. 需求挖掘
2. 提示工程
3. 系统安全
4. 用户接口
5. AI安全 & 对齐
6. LLMOps
7. 多模态功能
8. 环境交互性

生成式人工智能时代，研究机构可以研究的问题并没有减少，更多的只是内容上的更新，这也就要求学者敢于定义新任务，新场景，快速试错，并提出可能的解决方案

(1)你认为没有其他人会解决它。(2)你在这个主题上有独特的贡献能力。”

工具基础

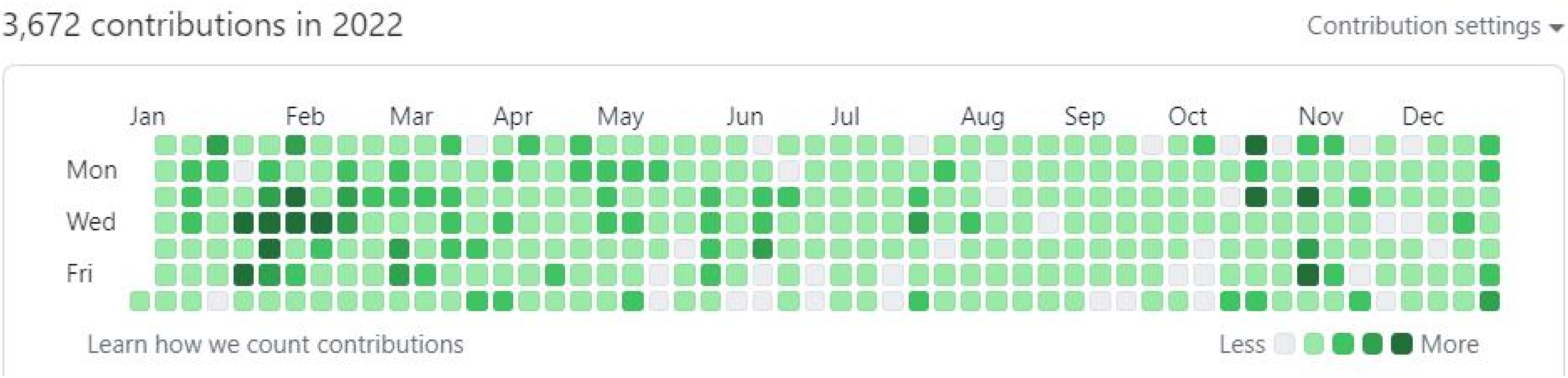
CS2916 大语言模型

飲水思源 愛國榮校

（该部分课件李学峰制作）

GitHub

- GitHub 是一种基于Web的代码托管平台
- GitHub使用Git作为版本控制系统
- GitHub的主要功能包括：
  - 代码托管： 开发者可以将他们的代码存储在GitHub上
  - 版本控制： GitHub使用Git进行版本控制，允许开发者跟踪代码的变化
  - 协作与社交： GitHub提供了一系列协作工具，如问题跟踪、合并请求、代码审查
  - 项目管理： GitHub允许用户创建项目、组织、团队

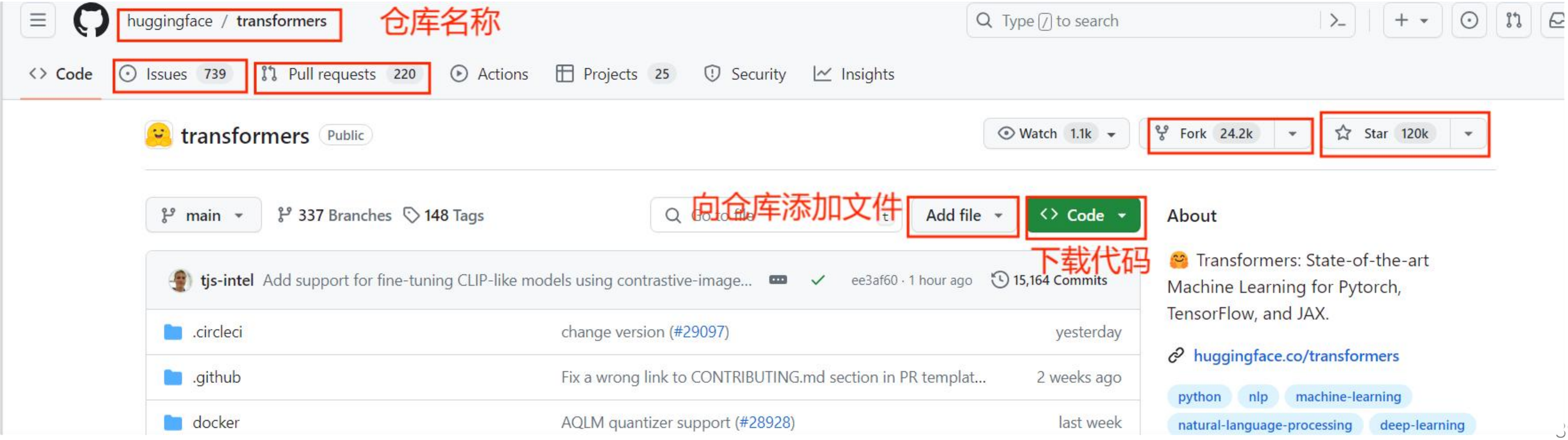


3,672 contributions in 2022

GitHub

□ 术语解释

- Repository：简称Repo，可以理解为“仓库”
- Issues： “问题”与项目开发者交流
- Star： “点赞”
- Fork： “拉分支”在自己账号下创建一个与原项目相同且独立的项目
- Pull Request：可以理解为“提交请求”，申请提交修改到目标项目
- Merge：“合并” ， 合并他人Pull Request



仓库名称

向仓库添加文件

下载代码

tjs-intel Add support for fine-tuning CLIP-like models using contrastive-image... ✓ ee3af60 · 1 hour ago 15,164 Commits

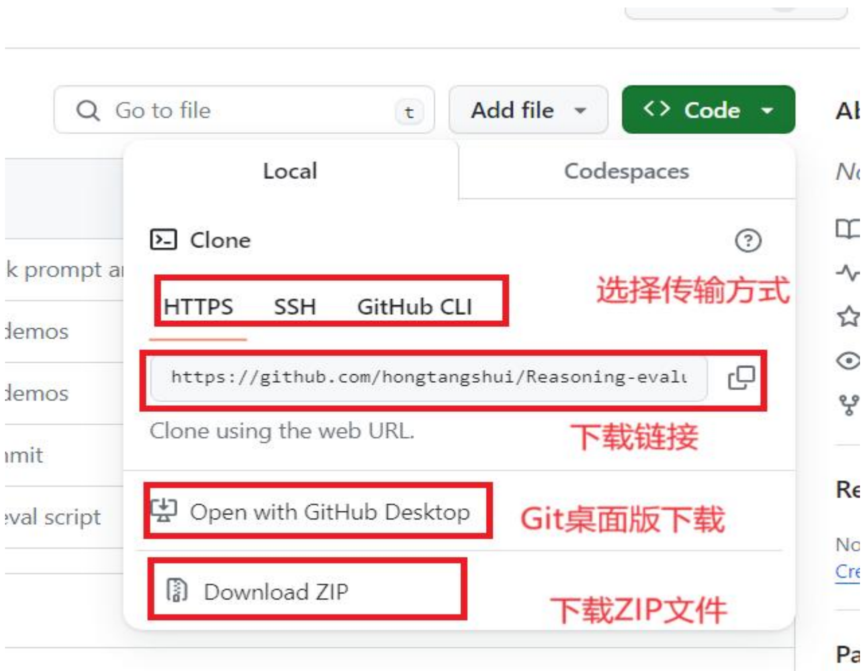
.circleci	change version (#29097)	yesterday
.github	Fix a wrong link to CONTRIBUTING.md section in PR templat...	2 weeks ago
docker	AQLM quantizer support (#28928)	last week

Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.

GitHub

□ 下载文件

- 下载：将repo中文件的快照作为zip下载
- Clone: 使用 Git 将存储库克隆到本地计算机
  - □ 使用命令行: git clone [下载链接]
  - □ 使用Git Desktop



https://github.com/hongtangshui/Reasoning-evalu

Clone using the web URL.

选择传输方式

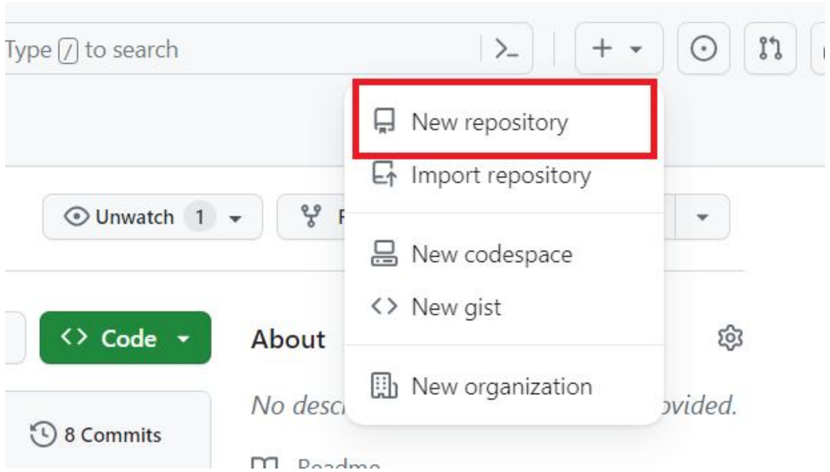
下载链接

Git桌面版下载

下载ZIP文件

□ Github创建新仓库

- Repo name
- 选择Public/Private
- 写Readme



GitHub

- Git是一个分布式版本控制系统，用于跟踪和管理开发项目中的源代码的变化
- 可以使用Git clone github上的代码，将本地代码同步到github上
- Git安装：
  - windows：https://git-scm.com/
  - Linux: apt-get install git
- 身份验证：git连接到github时，需要身份验证(HTTPS或者SSH)
- Git常见命令
  - git init: 将当前目录初始化为一个git仓库
  - git clone [下载链接]: 将一个远程项目clone到本地
  - git pull: 更新本地文件
  - git push: 将本地更新推送到远程仓库

GitHub

□ 教程

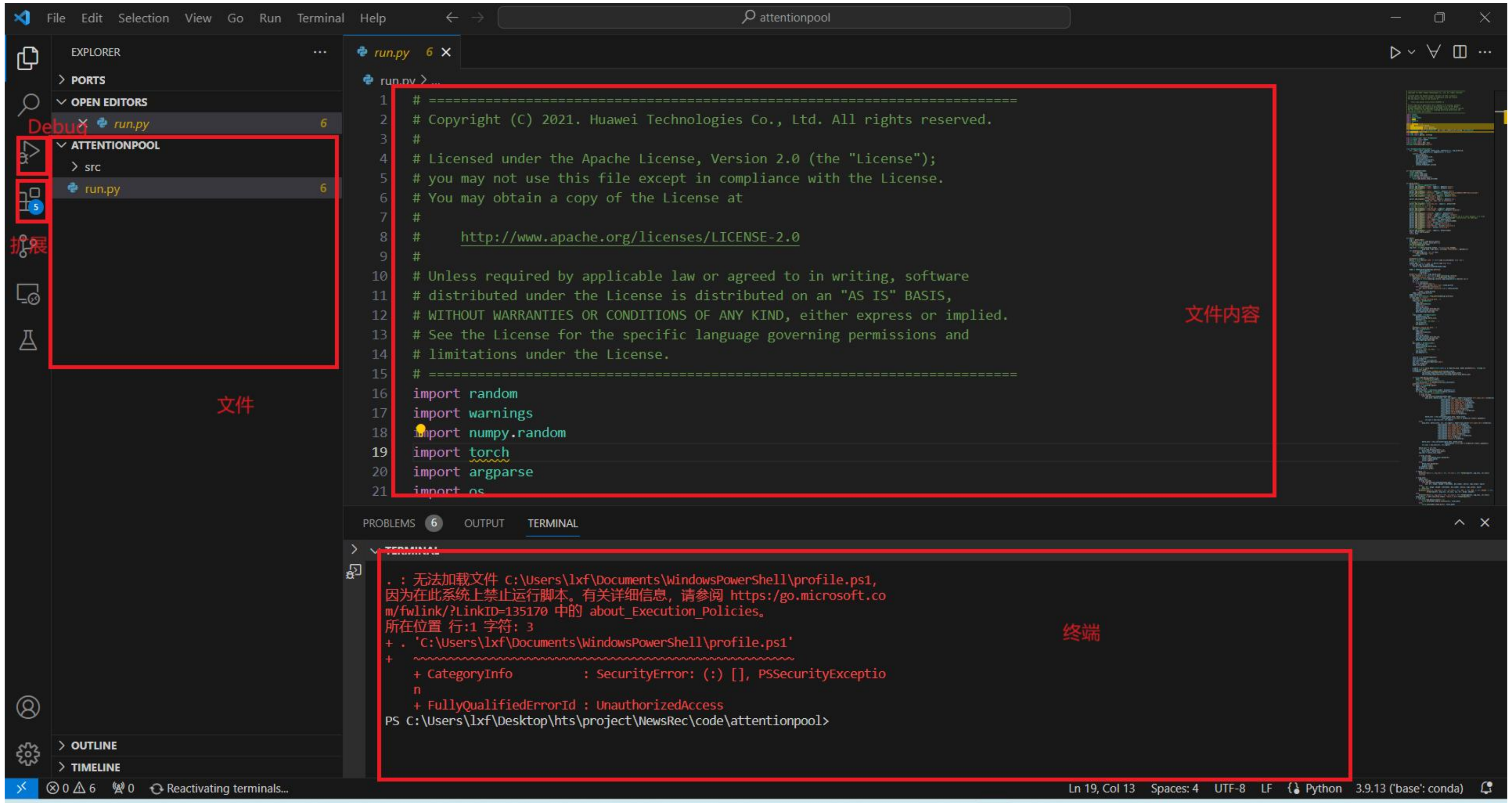
- 官方文档： https://docs.github.com/zh/get-started
- Github网站介绍: https://zhuanlan.zhihu.com/p/664195515
- Git命令介绍: https://zhuanlan.zhihu.com/p/369486197
- ChatGPT关于Github的答疑

VSCode

- □ Visual Studio Code的缩写，是一款由Microsoft开发的免费、开源的轻量级代码编辑器
  - ■ 轻量级
  - ■ 丰富的扩展支持：远程连接
  - ■ Git集成
  - ■ 代码编辑: 高亮，补全，代码折叠等
- □ 下载: Visual Studio Code - Code Editing. Redefined
- □ 设置: CTRL+，修改字体等

VSCode

- □ 新建/打开 文件/文件夹: 左上角File
- □ 打开终端: CTRL+SHIFT+Y
- □ 扩展：Python(运行python程序), Remote-SSH(远程连接)
- □ 运行代码: 可以F5键使用Vscode的Python扩展运行，也可终端命令运行



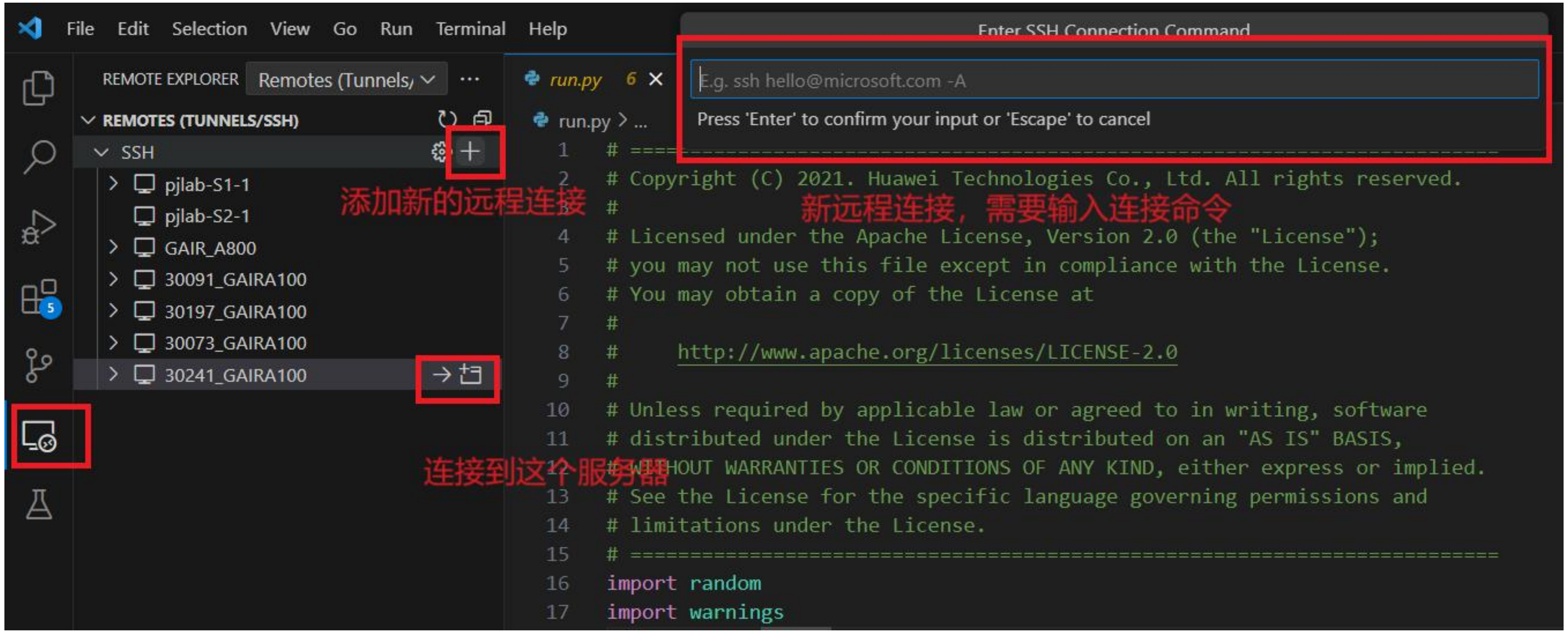
```
# ===========================================================================
# Copyright (C) 2021. Huawei Technologies Co., Ltd. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# ===========================================================================
import random
import warnings
import numpy.random
import torch
import argparse
import os
```

```
. : 无法加载文件 C:\Users\lxf\Documents\WindowsPowerShell\profile.ps1,
因为在此系统上禁止运行脚本。有关详细信息，请参阅 https:/go.microsoft.co
m/fwlink/?LinkID=135170 中的 about_Execution_Policies。
所在位置 行:1 字符: 3
+ . 'C:\Users\lxf\Documents\WindowsPowerShell\profile.ps1'
+   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityExceptio
    n
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\lxf\Desktop\hts\project\NewsRec\code\attentionpool>
```

VSCode

□ 远程连接

- 扩展安装Remote SSH
- 左边栏点击Remote Explorer
- 添加新的远程连接：输入连接命令，密码等



Enter SSH Connection Command

E.g. ssh hello@microsoft.com -A

Press 'Enter' to confirm your input or 'Escape' to cancel

添加新的远程连接

新远程连接，需要输入连接命令

连接到这个服务器

```
# ==========================================================================
# Copyright (C) 2021. Huawei Technologies Co., Ltd. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# ==========================================================================
import random
import warnings
```

VSCode

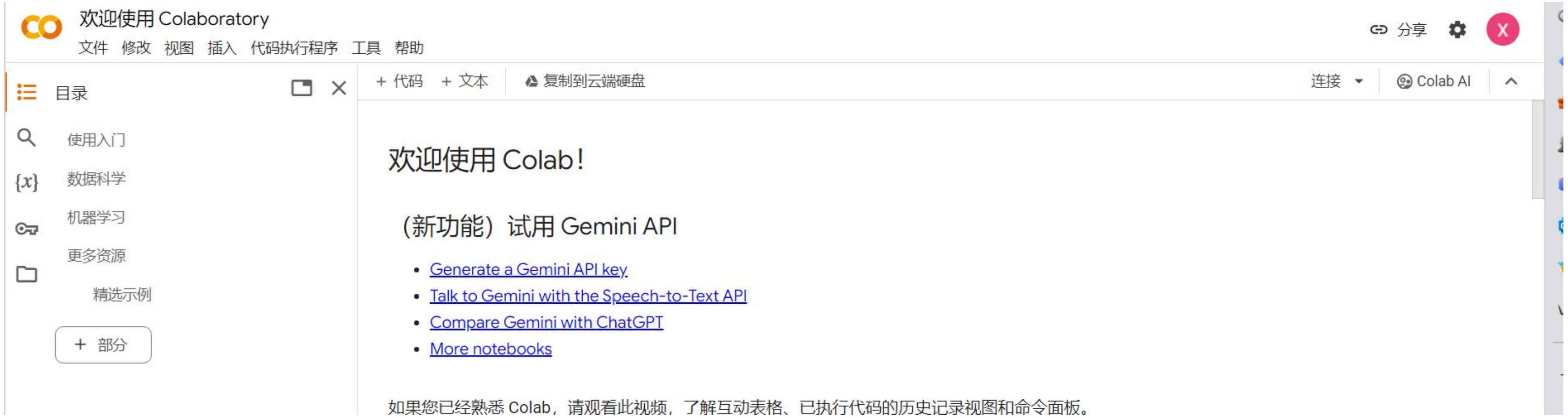
□ 教程

- 官方文档： https://code.visualstudio.com/Docs
- 远程连接: https://blog.csdn.net/zhaxun/article/details/120568402
- 运行: https://blog.csdn.net/zhangkai950121/article/details/117395333
- ChatGPT关于VScode的答疑: https://chat.openai.com/share/7ee10cff-689a-4ff4-a85d-10d63fb040b3

Google Colab

□ Google Colab

- https://colab.research.google.com/
- Google Colab是由Google提供的一种免费的云端Jupyter笔记本服务。它允许用户在云端运行和编写Python代码，而无需在本地计算机上安装任何软件
- 免费使用
- GPU支持
- 预装软件: 安装了常用python库和框架
- Jupyter支持: 基于jupyter notebook



欢迎使用 Colab！

（新功能）试用 Gemini API

- Generate a Gemini API key
- Talk to Gemini with the Speech-to-Text API
- Compare Gemini with ChatGPT
- More notebooks

如果您已经熟悉 Colab，请观看此视频，了解互动表格、已执行代码的历史记录视图和命令面板。

Google Colab

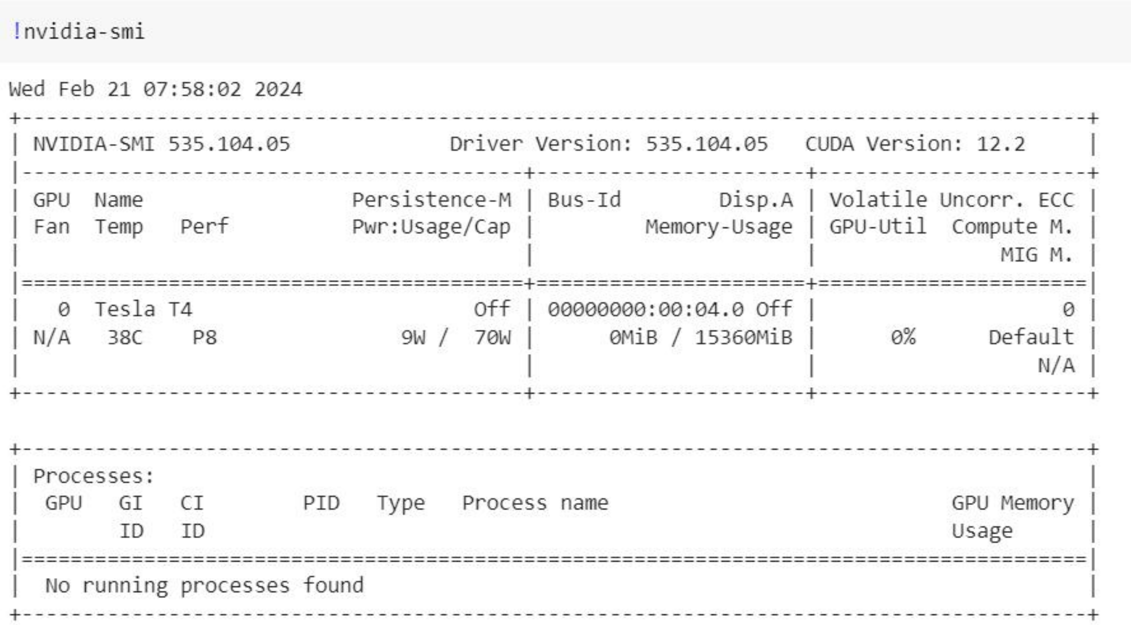
□ Jupyter Notebook

- 交互式计算，可以嵌入Markdown文本
- 使用
  - □ 一个jupyter notebook文件分为多个块，每个块为代码块或文本块
  - □ 代码块可以写python代码并运行，文本块写markdown文本
  - □ 运行代码块后，运行结果会保留在内存中(比如导入的包，写好的函数，计算好的变量)，再运行其他代码块时可以引用这些函数，变量
  - □ 使用Shift + enter执行当前块(代码块运行python，文本块渲染markdown)，并在下面创建一个新的代码块
  - □ 代码框中使用 !+命令 等价于在终端中输入命令
- Note: vscode中安装jupyter扩展之后，将文件扩展名改为ipynb也可以使用jupyter notebook

Google Colab

□ Google Colab设置

- 使用GPU：代码执行工具->更改运行时类型->硬件加速器
- !nvidia-smi 查看GPU



```
!nvidia-smi
```

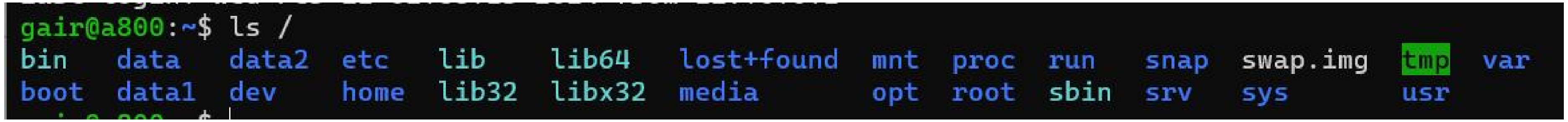
```
Wed Feb 21 07:58:02 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   38C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Google Colab

- 教程：
  - Google Colab官方教程：https://colab.research.google.com/#scrollTo=2fhs6GZ4qFMx
  - Jupyter Notebook文档: https://docs.jupyter.org/en/latest/
  - Jupyter Notebook教程: https://zhuanlan.zhihu.com/p/75402607
  - 本地使用Jupyter notebook: https://zhuanlan.zhihu.com/p/33105153

Linux

- Linux：开源操作系统
  - 远程服务器通常为Linux系统
- 服务器通常无GUI，只能命令行交互
- 系统目录结构
  - /home: 用户的主目录，在Linux中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的
  - /mnt: 挂载其他的文件系统
  - /root: 超级权限者的用户主目录
  - /usr：多应用程序和文件都放在这个目录下，类似与windows下的program files目录



```
gair@a800:~$ ls /
bin   data   data2  etc   lib    lib64   lost+found  mnt  proc  run   snap  swap.img  tmp  var
boot  data1  dev    home  lib32  libx32  media       opt  root  sbin  srv   sys       usr
```

Linux

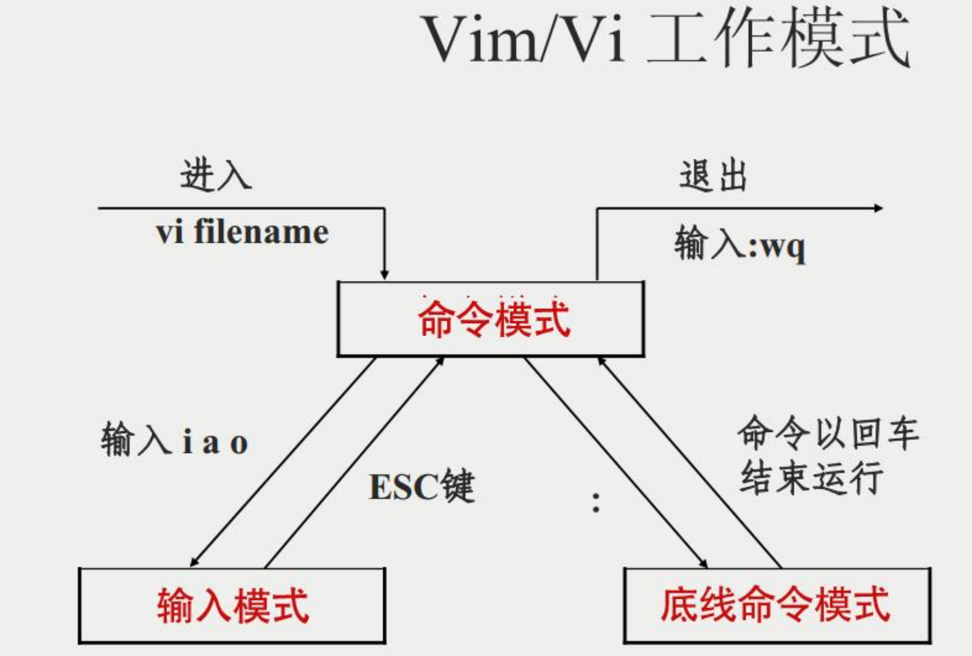
Linux常见命令：

- 使用man+命令 或者 命令 --help 可以查看命令的功能，参数，使用方法等
- 如 man cat; cat --help

命令	功能	命令	功能
ls	显示当前目录下子文件/目录	grep	查找文件中符合条件字符串
cd	切换目录	find	在指定目录下查找文件
pwd	显示当前目录	cat	显示文件内容
mkdir	创建目录	head	显示文件内容前n行
mv	移动文件/目录	tail	显示文件内容后n行
cp	复制文件/目录	df	列出文件系统整体磁盘使用量
rm	删除文件/目录	du	检查磁盘空间使用量
echo	打印字符串	chmod	改变文件权限

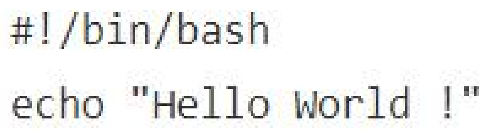
Linux

- □ Vim编辑器：
  - ■ 启动vim: vim+文件
  - ■ Vim分为三种模式
    - □ 命令模式: 刚启动vim，键盘输入被识别为命令而非输入字符;
    - □ 输入模式: 命令模式下按 i 进入输入模型，此时键盘输入为输入字符到文件
      - ■ 字符按键: 输入字符
      - ■ ENTER：回车
      - ■ DEL：删除
      - ■ ESC: 回到命令模式
    - □ 底线命令模型: 在命令模式下按 : 进入底线命令模式
      - ■ :wq —— 保存并退出
      - ■ :\xx —— 文件中查找xx



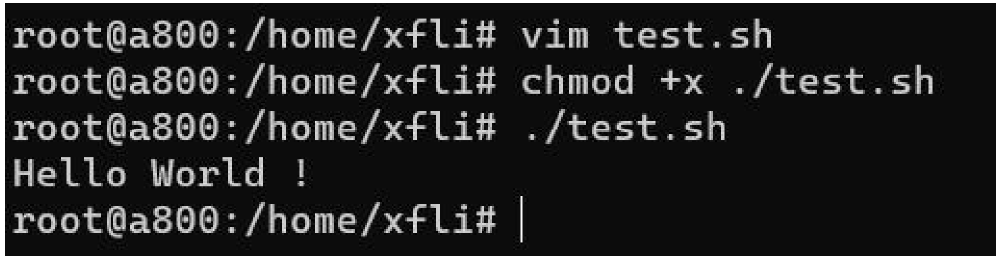
Linux

- □ Shell:
  - 命令解释器，与Linux内核交互的编程语言
  - Linux中常用的Shell是Bash
  - 编写一个bash脚本:
    - □ 创建一个文件: *vim test.sh*
    - □ 输入一些代码:



```
#!/bin/bash
echo "Hello World !"
```

- □ (第一行为指定解释器，第二行echo类似于C中printf)
- □ *chmod +x ./test.sh* 是脚本具有执行权限
- □ 运行: *./test.sh*



```
root@a800:/home/xfli# vim test.sh
root@a800:/home/xfli# chmod +x ./test.sh
root@a800:/home/xfli# ./test.sh
Hello World !
root@a800:/home/xfli# |
```

Linux

- 教程：
  - https://www.runoob.com/linux/linux-tutorial.html
  - https://www.w3cschool.cn/linux/linux-tutorial.html (包含命令，vim教程，shell教程等)
  - GPT答疑: https://chat.openai.com/share/f4d41c2e-8c9c-4293-8cb9-1f8c1f81a89f

Python

- □ Python:
  - 主流编程语言之一, AI领域使用最多的语言
  - 下载: https://www.python.org/
- □ Python虚拟环境:隔离项目中依赖关系和包的工具
  - 执行复杂的Python代码通常依赖一些写好的包
  - 不同项目需要的包不一样，且有可能需要同一个包的不同版本
  - 需要为每个项目单独构建一个虚拟环境，安装这个项目需要的包
- □ Anaconda:
  - 强大的Python包管理器，可以用于创建虚拟环境，并且为每个环境安装需要的依赖和包
  - Anaconda 安装: Free Download | Anaconda

Python

□ Anaconda使用:

- 创建虚拟环境: conda create –name myenv
- 激活虚拟环境: conda activate myenv
- 在虚拟环境中安装包(需要先激活虚拟环境): pip install numpy 或者 conda install numpy
- 运行python程序(先激活对应环境): python run.py
- 退出环境: conda deactivate
- 列出所有环境: conda env list
- 列出当前环境下所有包(先激活虚拟环境): pip list

Python

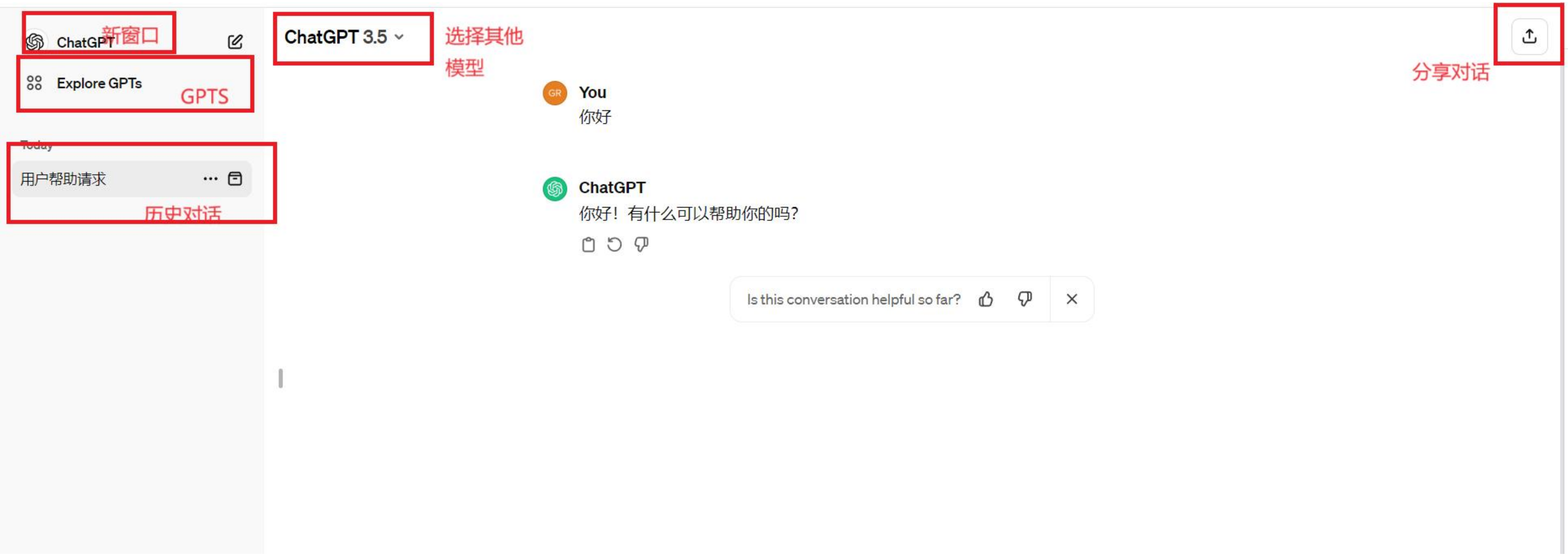
- Python基本语法
  - 使用缩进indent而非{}指示每条程序之间的关系，结尾无需分号;
  - 打印输入: print(“hello world !”)
  - 变量和数据类型:
    - Python不需要显示声明变量类型
    - 整数，浮点数，字符串，列表，元组，字典，集合
  - 控制流：与C基本相同
    - If elif else
    - for x in list:
  - 函数: 无需指名如何和返回值的类型，可以返回多个值
    - def add_numbers(a, b):
      - return a+b

Python

- □ 教程
  - Python: https://docs.python.org/zh-cn/3/tutorial/index.html
  - Conda安装:
    - □ Windows: https://blog.csdn.net/wyf2017/article/details/118676765
    - □ Linux: https://blog.csdn.net/fan18317517352/article/details/123035625
  - GPT答疑: https://chat.openai.com/share/37fa5858-bd99-430c-9606-efe97e3e69f4

ChatGPT使用

- https://chat.openai.com/



You
你好

ChatGPT
你好！有什么可以帮助你的吗？

ChatGPT使用

□ Prompt

- 应该包含以下内容
  - □ 角色或默认设置："你是一位聪明的人工智能助手"
  - □ 高层次目标的描述。
  - □ 子任务的详细项目列表（需要为每个子任务解释）
  - □ 演示/示例（对输出格式和指令遵循非常重要）
- *Note*:
  - □ *JSON*格式会导致性能较差（包括*GPT 4-1106*在内）
  - □ 演示应该多样且简洁。

ChatGPT使用

Anything as Prompting

你是一个中文人工智能助手，你需要仿照示例，根据给定的除示例外的所有法律生成一个包含题目、选项分析和答案的单项选择题。在生成单项选择题时，你必须遵守以下几个原则：

题目构成 1. 题目由题目描述和4个选项构成

题目描述 2. 单项选择题的题目描述需要合理

题目生成的整体限制 3. 尽可能根据除示例外的所有法律生成题目，避免使用单条法律生成题目

题目选项
4. 在生成4个选项时，结合题目描述与除示例外的所有法律，首先设计1个正确答案的选项，然后再设计3个错误的选项，接着这4个选项以随机的顺序排列
5. 选项互有差异，避免选项之间的明显重复或相似性
6. 在设计选项时，不要使得某些选项明显不可能是正确答案
7. 每个选项需要和题目描述相关
8. 每个选项需要前后内容一致
9. 不能直接从给定的法律中复制文本作为选项内容，需要结合给定的法律生成合理的选项

生成顺序 10. 依次生成题目、选项分析和答案

选项分析 11. 选项分析是结合题目与除示例外的所有法律，对每个选项进行分析

答案 12. 选项分析中的正确答案是最终答案

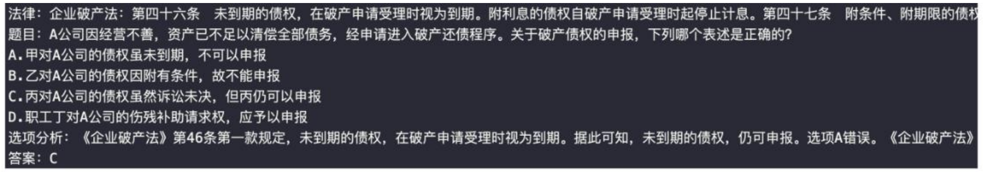
以下是1个示例：
示例：
{example}

让我们一步一步思考，参考示例并结合给定法律"{input_law}"{action}，依次生成下面内容：

题目：

选项分析：

答案：



```
法律：企业破产法：第四十六条　未到期的债权，在破产申请受理时视为到期。附利息的债权自破产申请受理时起停止计息。第四十七条　附条件、附期限的债权
题目：A公司因经营不善，资产已不足以清偿全部债务，经申请进入破产还债程序。关于破产债权的申报，下列哪个表述是正确的?
A.甲对A公司的债权虽未到期，不可以申报
B.乙对A公司的债权因附有条件，故不能申报
C.丙对A公司的债权虽然诉讼未决，但丙仍可以申报
D.职工丁对A公司的伤残补助请求权，应予以申报
选项分析：《企业破产法》第46条第一款规定，未到期的债权，在破产申请受理时视为到期。据此可知，未到期的债权，仍可申报。选项A错误。《企业破产法》
答案：C
```

中华人民共和国河道管理条例规定：第十条　河道的整治与建设，应当服从流域综合规划，符合国家规定的防洪标准、通航标准和其他有关技术要求，维护堤防安全，保持河势稳定和行洪、航运通畅。第十一条　修建开发水利……

设计一个法律情景/针对给定法律中的某个概念

谢谢各位！