预训练与词向量 从入门到放弃

NLP研发部
柴业坤
2019-01-09

JDAI Research

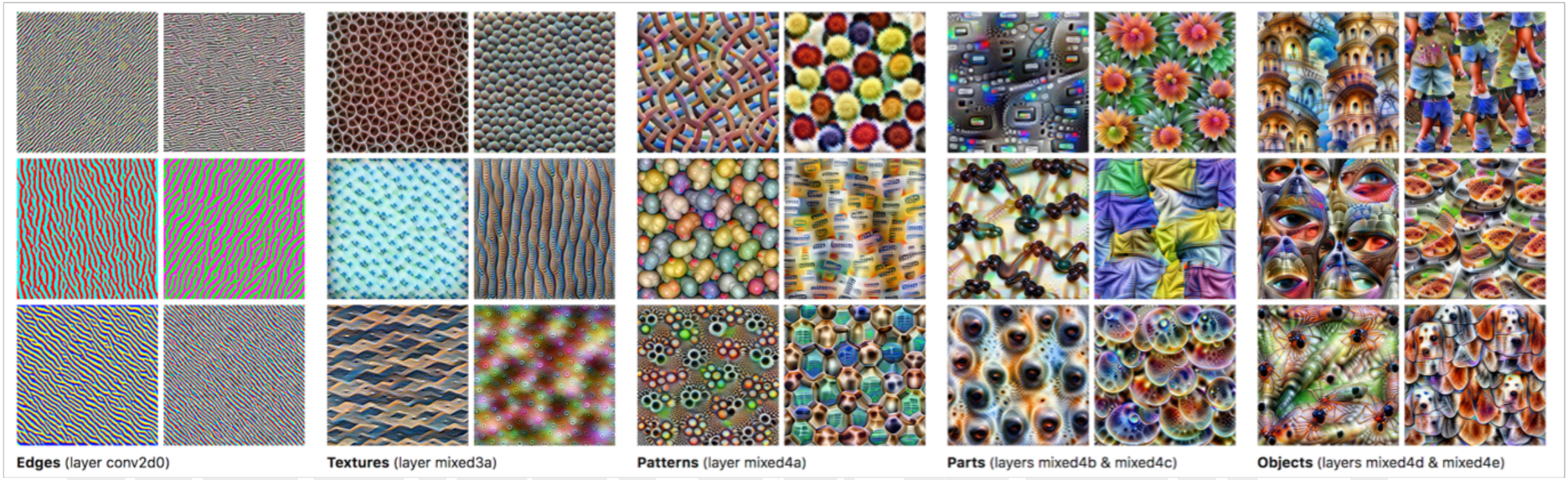
2018年度 NLP 里程碑事件

TOP1：预训练模型

JDAI Research

❑ Why 预训练？

- **CV performed extremely well with pretraining on ImageNet**



GoogLeNet 在不同层预训练学习到的特征。

NLP epistemology

Everything is classification in NLP .

—— Yekun

JDAI Research

NLP ∈ 分类

- 句子级别：情感分析(单模态)，意图分析，domain分类, …
- 1+句子级别：相似度，文本推理，…
- 1- 句子级别：槽值提取，命名实体识别，中文分词，词法/句法分析，指代消歧,…

什么是 词表示

JD AI Research

(很久很久以前) 如何表示一个词？

词表示

分布假设：语言学中具有相似分布的item具有相似的意义。

- ✓ 索引 1，2，3， 4，…
- ✓ 独热编码. 0000010000
- ✓ 词频（count / frequency）

- ✓ TF-IDF
- ✓ PMI,
- ✓ …

(很久很久以后) 如何较好地表示词(们)？

语言模型

预测当前句子的概率

离散n-gram

- S = #你吃饭了么
- P(S) = P(你|#) P(吃|#你) P(饭|#你吃) P(了| |#你吃饭) P(么| |#你吃饭了)
  ≈ P(你|#) P(吃|#你) P(饭|#你吃) P(了|你吃饭) P(么|吃饭了)

- 目标： $P(S) = P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{o\ \ldots,\ w_{i\ \_\ n\ +\ 1})}$

- 强马尔可夫假设：每个词 的概率只取决于前面几个词，
  而与更前面的词无关。 $P(S) \approx \prod_{i=1}^{n} P(w_0 | w_{i\ \_\ 1, \ldots,\ wi\ \_\ n\ +\ 1})$

  - Unigram：词频
  - Bigram：Wi只取决于Wi-1
  - Trigram：Wi只取决于Wi-2, Wi-1

✓ P(么|饭了) = $C(\text{饭了么}) / C(\text{饭了})$

这么做有什么问题？

- 无法处理训练集中没有出现的词X， Count(X)=0；
- 需要大规模训练预料保证n-gram不同的组合出现频率；

思路：从原始概率分布中预留一部分概率给未出现的词。

解决方案：

- 加一平滑
- 加α 平滑
- 补偿法
- 插值法
- KN平滑

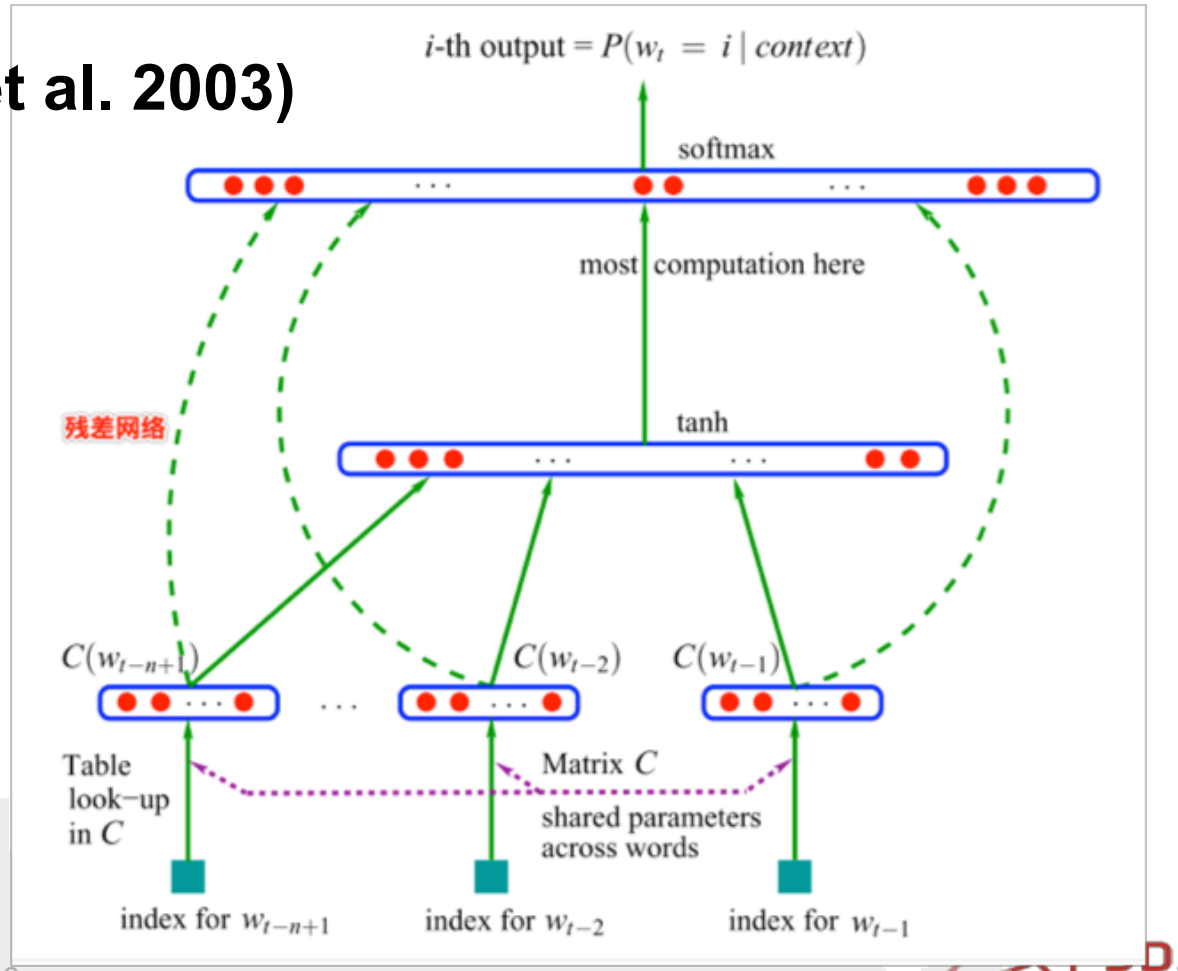
连续n-gram

想法：依赖词内部之间的**相似性**，估计训练集中未出现的词。

✓NPLM 神经概率语言模型(Bengio et al. 2003)

同时学习词的分布表征和句子的联合概率分布。

- 输入：n-1 个context word
- 输出：下一个词的概率分布
- 模型：线性映射层，✚ DNN隐藏层
- Optionally, 将映射的词直接喂给输出层



问题：

- 预先定义好的上下文长度n
- 简单的网络结构（现在看来）
- 向量空间里一个词映射成一个点
- 不能处理多义词问题

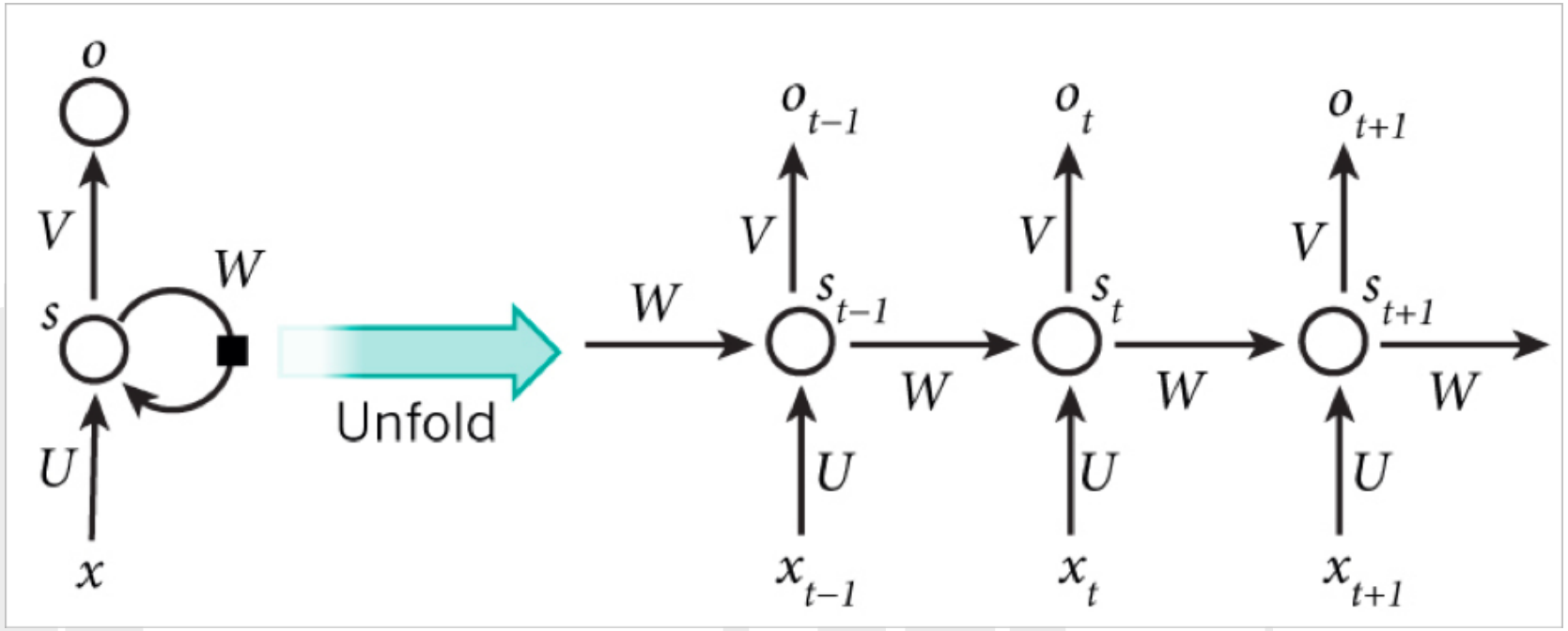
连续n-gram

✓RNNLM (Mikolov et al. 2010)

- 输入：独热编码（维度|V|）
- 输出：下一个词的概率分布
- 模型：Simple RNNs
- 参数：
  - 向量矩阵 E
  - 前向网络参数矩阵 V, U
  - Recurrent 矩阵：W

解决的问题：

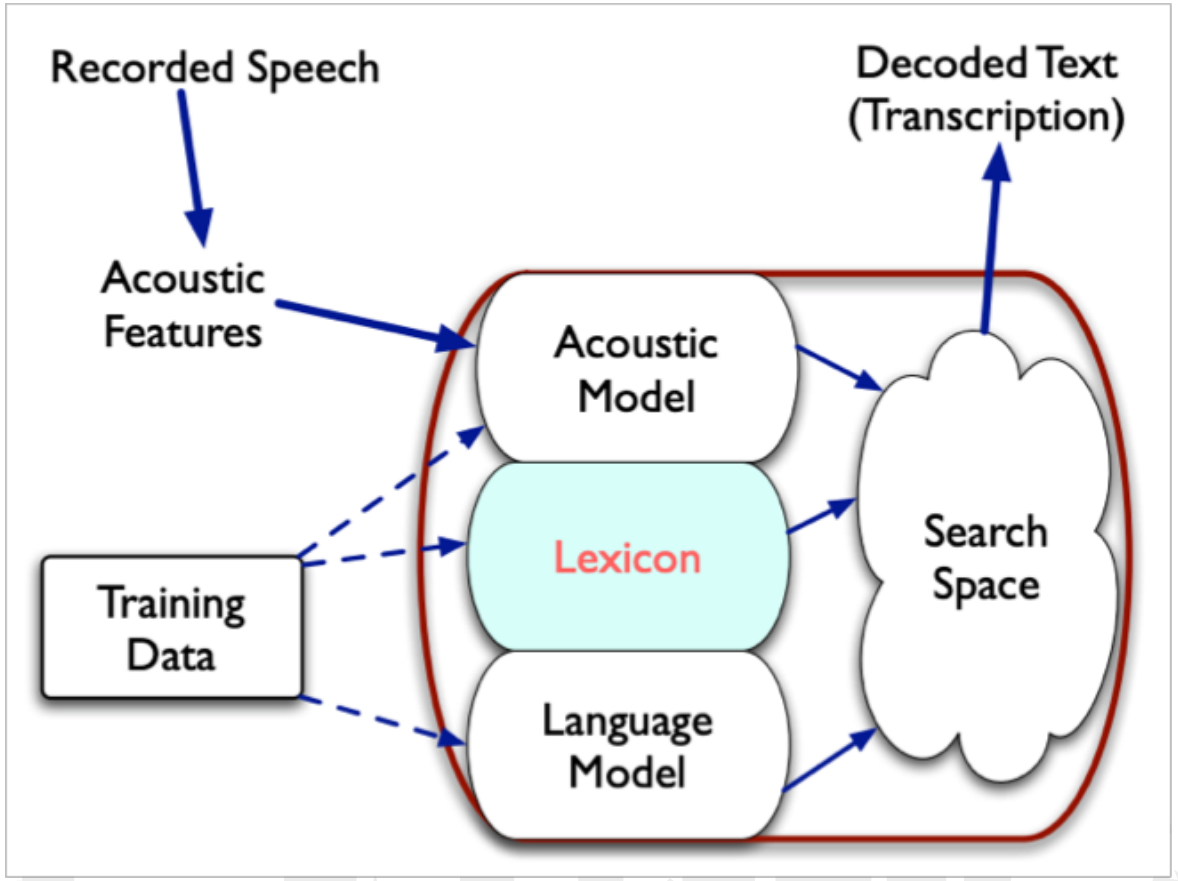
- 预先定义好的上下文长度n



(Image source: A. Lopez)

语言模型的用途

- ASR
- MT
- Spelling correction
- …



较好的表示

理论：You shall know a word by the company it keeps 。 (John Rupert Firth, 1957).

- ✓ **固有缺点：**
  - ✓ 不能区分词序，无法表示惯用语

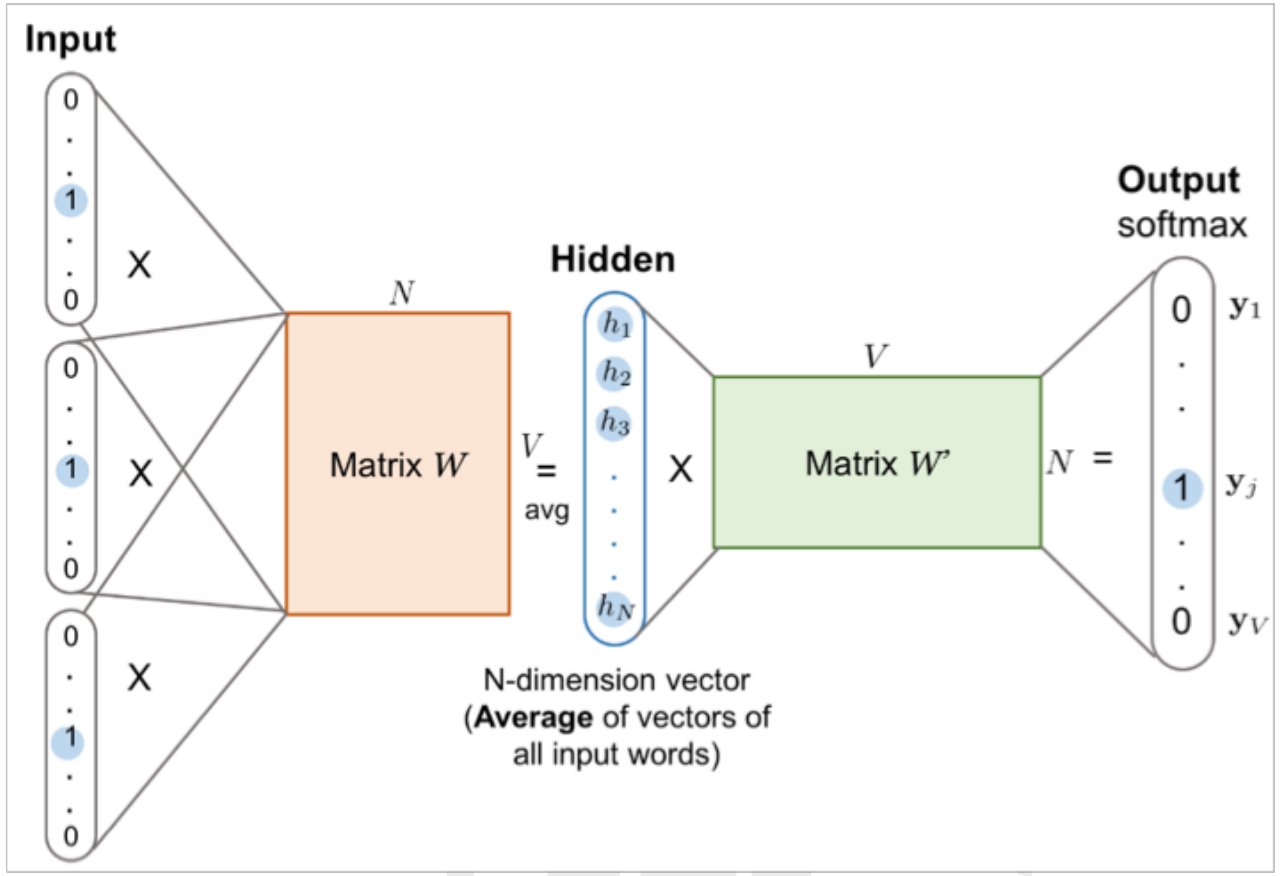
✓向量空间词表征

- ✓ 目标：捕捉**更细粒度**的**语言学**规律
- ✓ 我的分类：
  - ✓ 基于神经网络模型
    - ✓ **Word2Vec** (Google 2013)
    - ✓ Collobert and Weston embeddings
    - ✓ HLBL embeddings
    - ✓ …
    - ✓ **Fasttext** (Facebook 2017)
  - ✓ 基于矩阵分解模型
    - ✓ LSA
    - ✓ **GloVe** (Stanford 2014)
- ✓ 优势

较好的表示

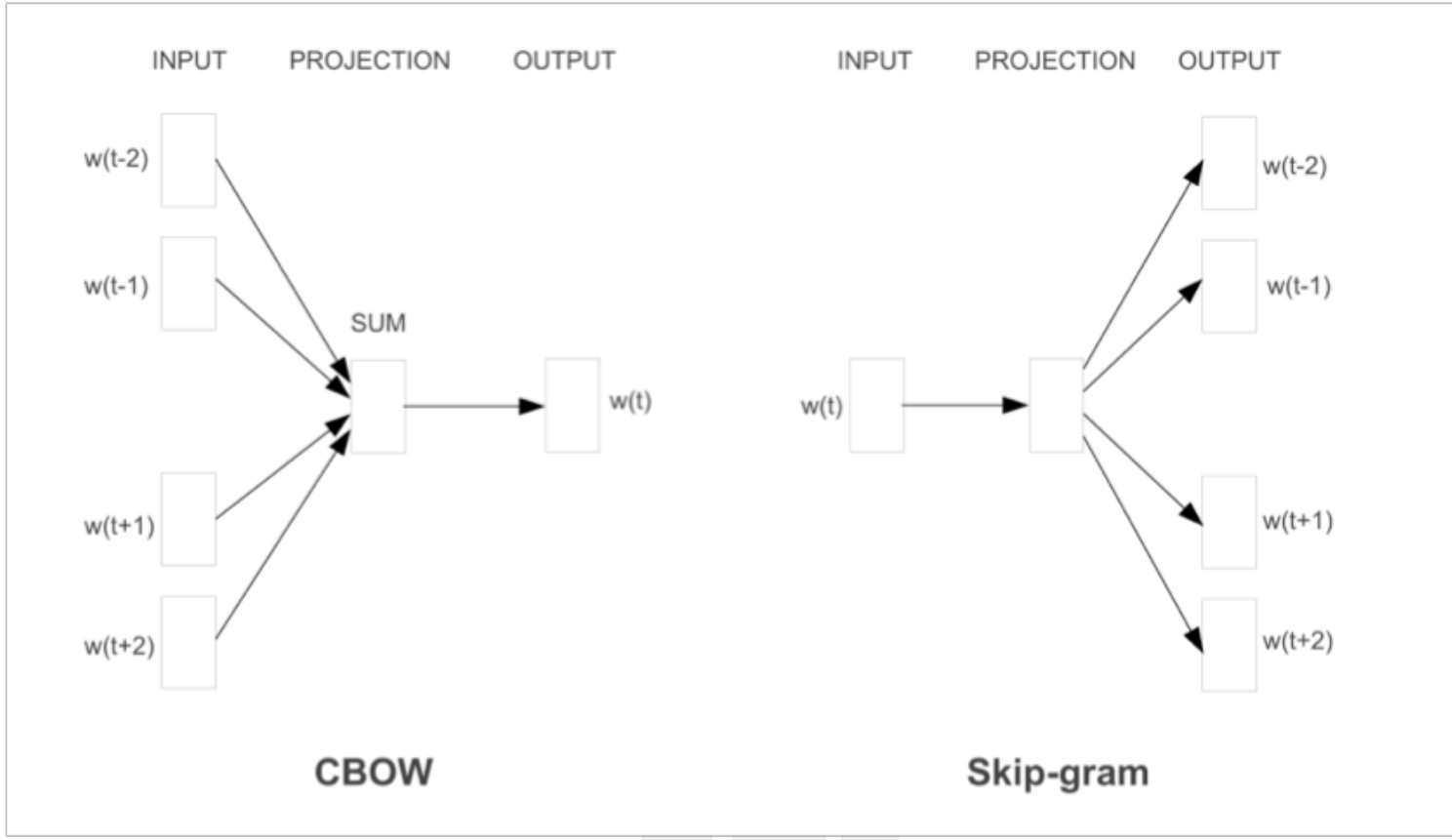
- ✓ **Word2Vec (Google 2013)**
  - ✓ 连续词袋模型 CBOW

- ✓ 想法：基于上下文的词预测当前词。
- ✓ 模型结构：线性层（NPLM去掉非线性层）
- ✓ 所有词被投射到相同位置（取算术平均）。
- ✓ 和前面讲的NPLM相比，算力更有效。



较好的表示

- ✓ **Word2Vec (Google 2013)**
  - ✓ Skip-gram
- ✓ 想法：利用当前的词预测在一定窗口大小内上下文的词。
- ✓ 简单的加法可以产生有意义的词
  - ✓ vec(“Russia”) + vec(“river”) ≈vec(“Volga River”)
  - ✓ vec(“Germany”) + vec(“capital”) ≈ vec(“Berlin”).

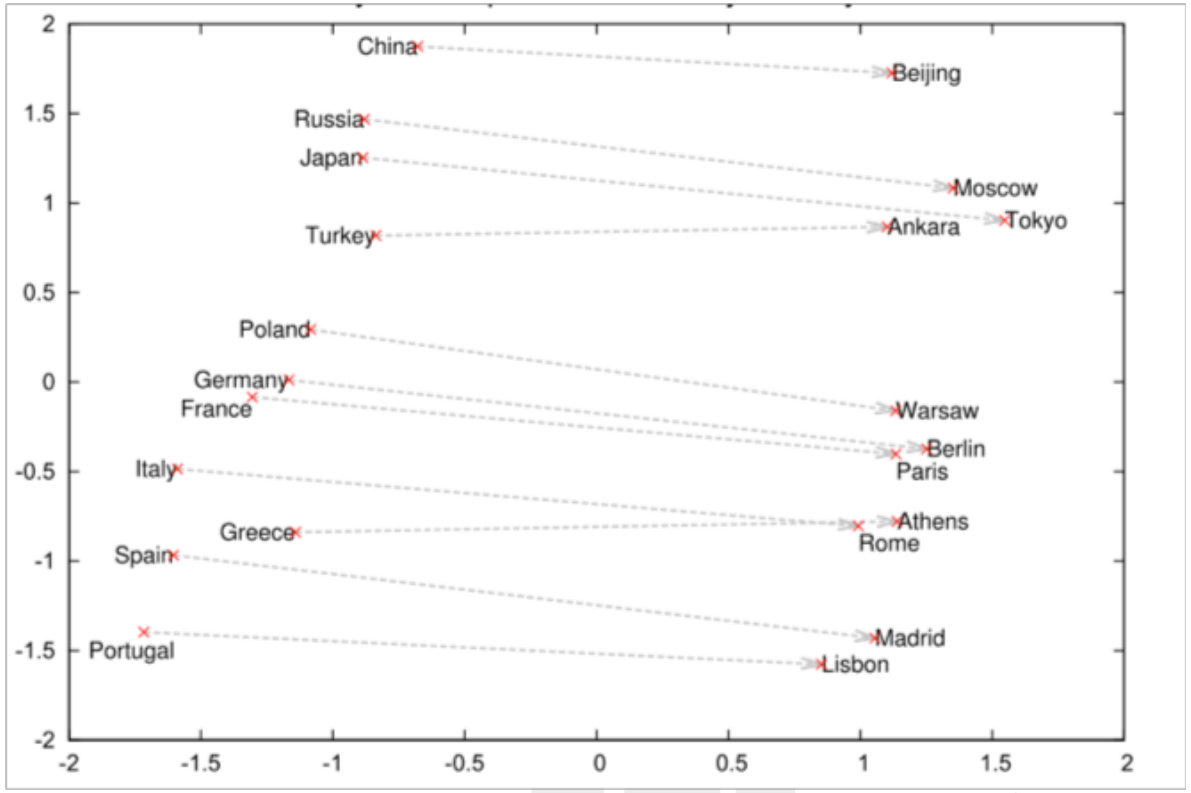


- ➢ **问题 1.不能表示常用短语**
- ✓ **Solution：将常用短语作为一个iterm训练**
- ✓ vec(“Montreal Canadiens”) - vec(“Montreal”) + vec(“Toronto”) = vec(“Toronto Maple Leafs”).
- ➢ **问题 2. softmax层非常大的维度计算**
- ✓ **Solution：**层次softmax(Morin and Bengio, 2005), 负采样(Mikolov et al. 2013)

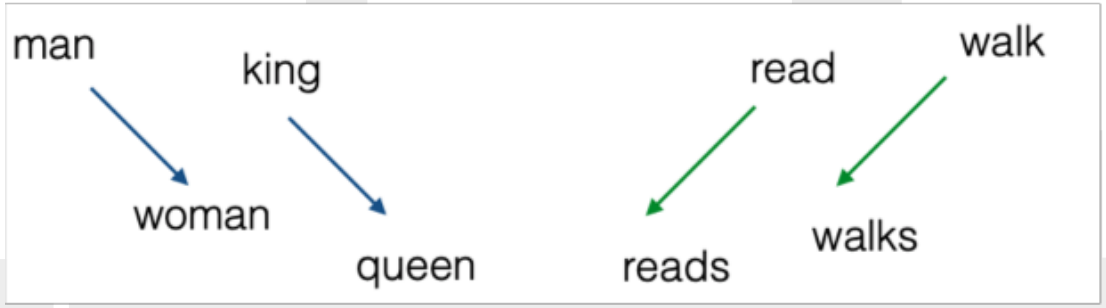
较好的表示

✓ **Word2Vec (Google 2013)**

- 捕捉到类推关系



- 捕捉到语义和词法信息



较好的表示

✓ GloVe (Google 2013)

- 思想：利用统计语言信息，在非0元素上构建词词共现矩阵。
- 改进：
  - 利用**共现概率的比率**而不是共现概率本身来捕捉语义。全局向量对于两个词i，j 之间的关系，基于第三个词 k （比率）进行建模。
- 两步：
  - 构建词词共现矩阵
  - 全局矩阵分解

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

较好的表示

✓ Fasttext (Facebook 2013)

- **遗留问题**：忽略词内部之间的关系，对于词法形态变化大的语言无法有效建模，例如土耳其🇹🇷语言语和芬兰🇫🇮
- 改进Skip-gram：
  - 考虑小于字级别的表示（包括本身），建立特征集合
  - 求和

例如： <where>

- 字符级别:<wh, whe, her, ere, re>
- 本身:<where>

半监督信息抽取 / 序列标注，非正式用语词向量评估

Discovering **spelling variants** on **Urban Dictionary**

JDAI Research

Discovering spelling variants on Urban Dictionary

- **Previous work**: Saphra, N. and Lopez, A., 2016. Evaluating Informal-Domain Word Representations With Urban Dictionary. arXiv preprint arXiv:1606.08270.
- Motivation 动机
- Challenge 挑战
- Task 任务描述
- Methods 方法
- Results 结果
- Contribution 论文贡献

Motivation

- 当前词表征评估方法：穷举**外部评价**方法，即 在传统NLP任务上进行穷举法测试（e.g. 情感分析，命名实体识别，语义角色标注）。
  - **Problems**：假设所训练词向量有3种(skip-gram, Fasttext, Glove)，每种词向量设置3组超参数(max vocab, embedding size, window size)，每个超参数有3个候选值，则有 3^3 组排列组合。同时，假设在以上三个任务上进行验证，每个任务一个模型，3组超参数每组3个取值，对所有词向量进行网格搜索共有 3^6 组排列组合。
  - **缺陷**：需要大量GPU计算资源 和 人力资源， 调参时间。
- Solution：**内部评价**方法
  - Idea: 通过衡量词向量在**词表征高维空间**的分布，在**整体上**评估词表征的质量, e.g. word similarity, analogy (Kevy et al. 2015).
  - Requirement: 含有特定关系的词对的数据集。
  - 当前存在的数据集均为formal domain：WordSim353 (Finkelstein et al., 2001), SimLex-999 (Hill et al., 2015), MSR's analogy dataset (Mikolov et al., 2013b)
  - 没有大规模的informal domain (主要为社交domain, i.e. Twitter, 微博，论坛，聊天，贴吧)关系词对数据集。(Saphra & Lopez, 2016) 提出一个小规模(700+）的基于 lexico-syntactic 规则的 spelling variant（拼写变体）数据集。

Challenge

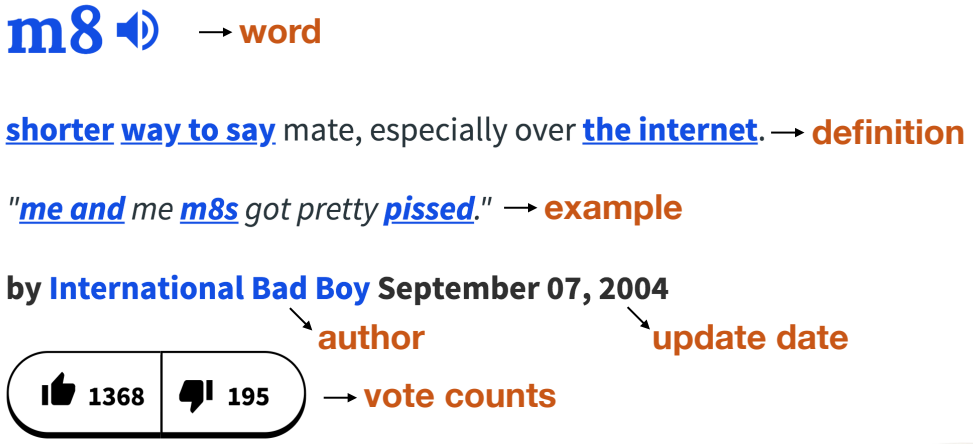
NO EXISTING LARGE SCALE Dataset for informal Domain !

Task

- 1. 信息抽取
  - 抽取 spelling variants 二元组（词对）
  - 数据源： Urban Dictionary（保证数据独立性）
- 2. 词向量评估
  - 使用提取的spelling variants关系数据集进行词向量内在评估
- 3. 比较
  - 与传统外部评价方法进行相关性比较（外部评价基于Twitter话题预测任务）

1.1 数据爬取

- 数据来源：英语俚语词典 -> 城市词典 Urban Dictionary （ http://urbandictionary.com ）



defid	word	definition
19	Hazy	A guys state of mind after he sees the girl of his dreams...He just can't believe it.
21	hork	to steal
22	hecka	see synonyms at hella.
23	hella	see synonyms at hecka.
24	hecka	a description of an excess of emotion, objects, or action.First used by Horatio Alger in 1902 in the streets of
30	twomp	a twenty dollar bill. Jackson's on it."tw" + "awh" + "mp".
32	ducket	a one dollar bill. $1.equivalent to one hundred pennies or twenty nickels or ten dimes or four quarters.
33	beefcake	to become overweight or buff
35	mad	a multi-functional word: very/a lot/hard/etc. Accentuates any verb.
42	clap	a case of gonorrhea
44	cob	a sharp poke or goose in the anus
47	puke	to vomit
48	folks	p. noun: People, not necessarily related, to whom you are close.
49	holla	v. to contact or communicate with, esp. after a long absence of communication
50	dog	n. friend of the same sex, usually male. Derived from the members of the Dogg Pound. pl.: dogs
54	raunchy	distasteful, obscene, and or just plain gross
55	ENERGY	can be converted from one form to another, but it cannot be created or destroyed.
61	Rental	means of transportaion that damage is totally irrelevant
62	cassette	ancient form of storring audio
66	raw	syn pure, unadulterated, hard core, serious, no kidding, no shit
69	AIM	digital for of communication

1.2 信息抽取

半结构化数据　关系抽取

geet	another word for git
tanks	[Profanese] variation of “thanks".
fishy	a kid word for "fish"

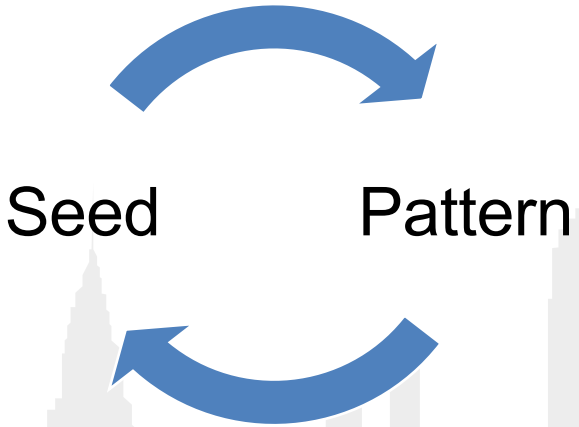
Urban Dictionary

word	variant
geet	git
tanks	thanks
fishy	fish

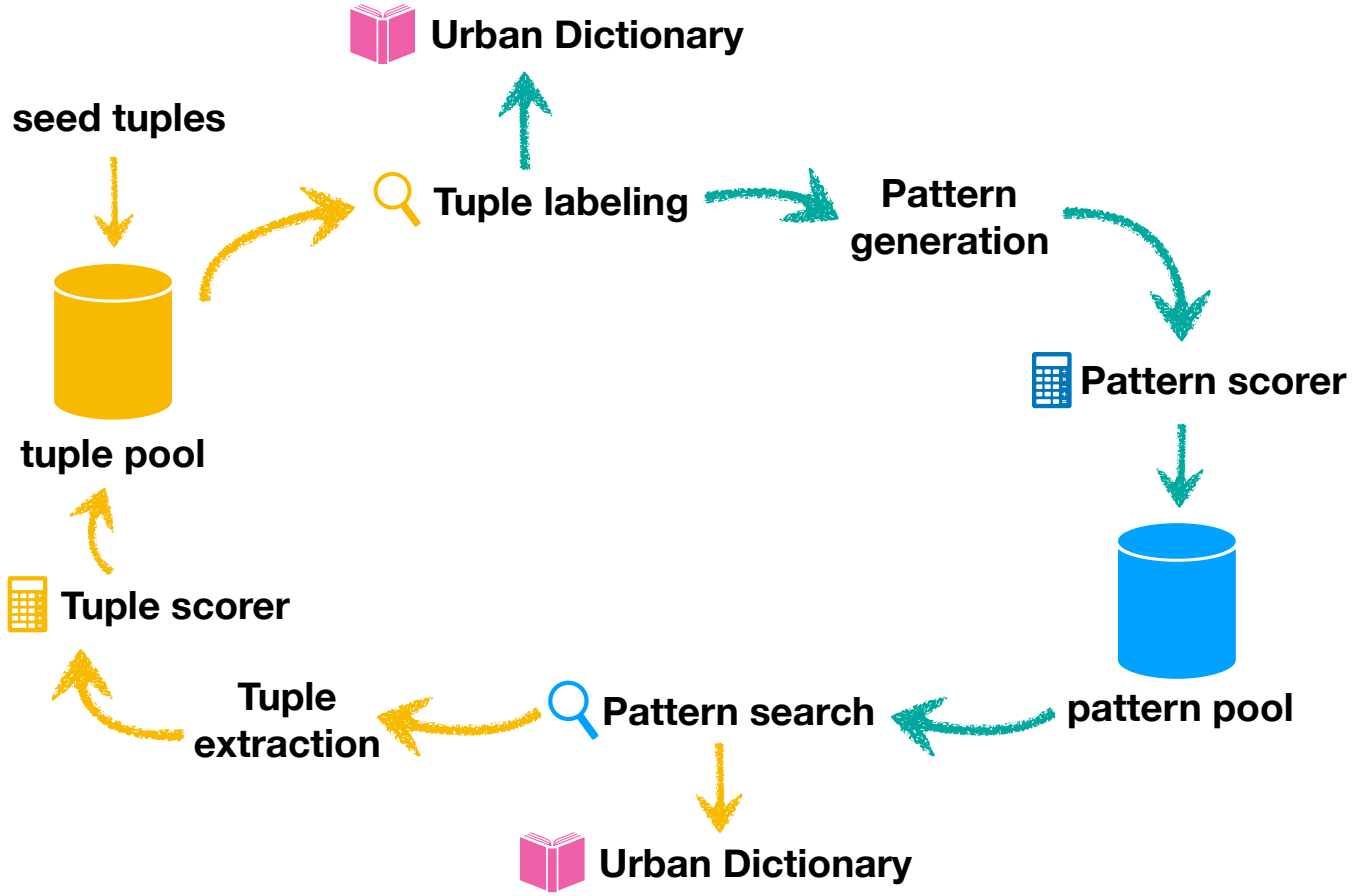
extracted variant pairs

- Challenge：**NO LABEL**！
- Only one previous work： Lexico-syntactic surface rule-based approach
- My solution: **半监督学习**
  - 1. Weakly-supervised pattern-based bootstrapping
  - 2. Self-training based CRF tagging

Approach 1: Weakly-supervised pattern-based bootstrapping



pipeline



- **8 iterations**
- 对每轮提取的spelling variants结果进行 **随机抽样**，检验 **accuracy**

Algorithm 2: Weakly-supervised pattern-based bootstrapping algorithm

Input : Unlabeled *Urban Dictionary* corpus (i.e. word entries $\mathbb{W}$ and corresponding definition sentences $\mathbb{D}$), seed tuples S (i.e. a small set of spelling variant tuples discovered by the baseline of pattern-based extraction), the threshold of pattern score pat_{min}, the threshold of tuple score tup_{min}

Output: A set of variant spelling pairs.

```
Initialize tuple pool with given initial seeds;
Iteration count = 0;
while Iteration count < Max Iteration do
    Find and label the occurrence of tuples in tuple pool from 𝕎, 𝔻 ;
    Generating candidate patterns based on the context lexico-syntactic
     information;
    Score each candidate pattern;
    if the confidence of candidate pattern > pat_min then
        Add into pattern pool;
    else
        Remove noise patterns;
    end
    Searching matched candidate tuples from 𝕎, 𝔻, using patterns in pattern
     pool;
    Score each candidate tuple;
    if the score of candidate tuple > tup_min then
        Append into tuple pool;
    else
        Remove noise tuples;
    end
    Iteration count += 1;
end
```

Approach 2: self-training based CRF tagging

半监督序列标注任务

- 1. 人工标注1500条序列标注格式数据，作为golden data；
- 2. 特征工程，使用当前标注数据训练CRF模型（trick：交叉验证，随机搜索）；
- 3. 用训练好的CRF对所有未标记数据进行预测，筛选出置信度大于某个阈值(85% / 90% here)的数据作为silver data；
- 将silver data合并到golden data；
- GoTo step 2，直到满足停止条件。

Algorithm 4: The process of self-training based CRF system

Given: Small amount of labeled word sequence X, a large amount of unlabeled data U, positive tag I, negative tag O

Output: The predicted label sequence Y with the highest conditional probability $P(Y|X)$

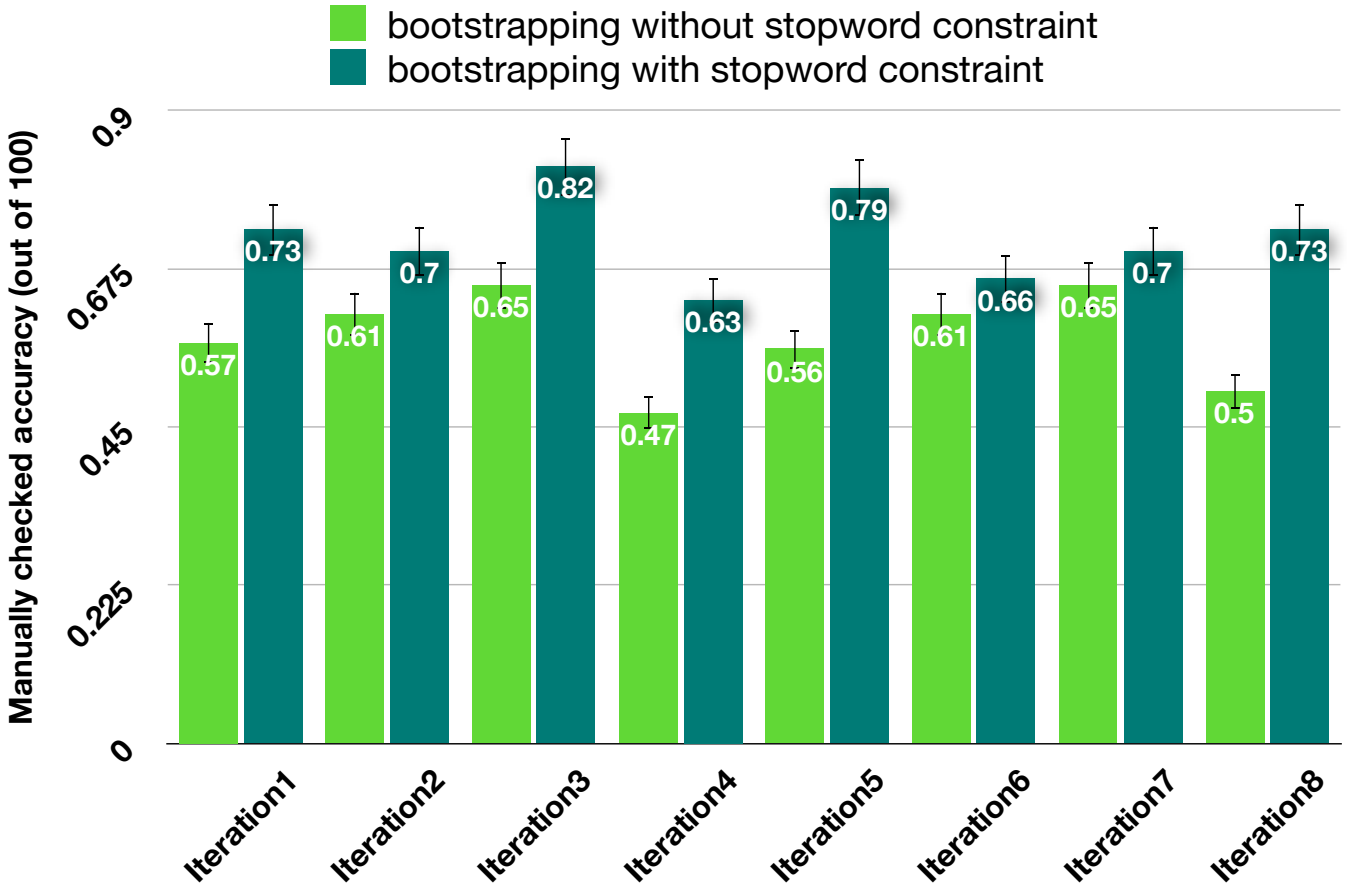
```
# load gold labeled data for the first iteration;
X = load_labeled_data_X() ;
while Iteration number <= max iteration number do
    F_x = feature_generation(X);
    crf = train_CRF(F_x);
    SilverData = [ ];
    for each sample u ← 0 to length(U) do
        y_u = crf.predict(u);
        if y_u == 'I' && tag_u > threshold then
            y_u = 'I';
            SilverData ← SilverData + (u, y_u)
        else
            y_u = O;
        end
        X ← X + SilverData;
    end
end
```

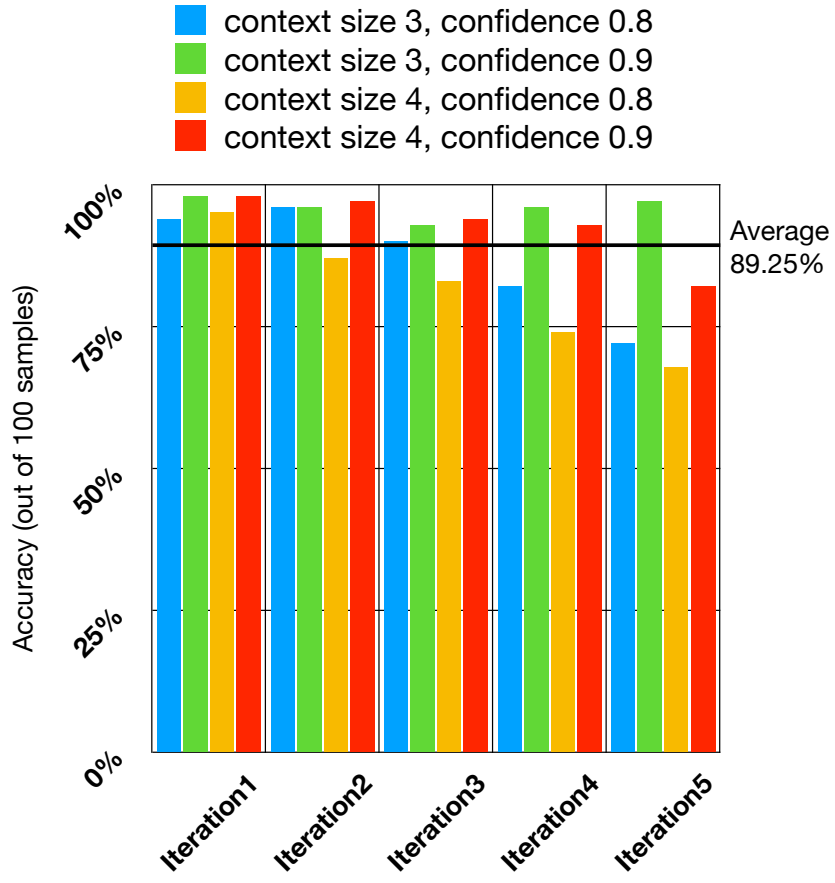
y=I top features		y=O top features	
Weight?	Feature	Weight?	Feature
+1.916	-2:lemma_=spelling	+3.885	pos_=PUNCT
+1.501	-1:word.lower=word	+2.646	word.lower=it
+1.474	-1:head.text=mispelling	+1.845	-3:head.text=combination
+1.236	-1:word.lower=spelling	+1.221	-3:head.text=in
+1.160	-1:tag_=``	+1.060	dep_=det
+1.148	head_text=for	+0.945	-1:pos_=ADJ
+1.133	-3:word.lower=way	+0.734	dep_=prep
+1.127	-3:lemma_=way	+0.718	lemma_=word
+1.110	head_text=saying	*… 122 more positive …*	
+1.055	-2:head.text=misspelling	*… 170 more negative …*	
+1.046	-3:lemma_=spelling	-0.724	-1:lemma_=word
+1.025	head_text=of	-0.776	-3:head.text=mispelling
+0.983	head_text=say	-0.789	-2:word.lower=word
+0.932	-3:head.text=misspelling	-0.820	-3:head.text=variation
+0.912	-2:lemma_=spell	-0.866	-2:head.text=for
+0.887	-3:head.text=version	-0.871	-2:head.text=of
+0.871	-2:head.text=of	-0.887	-3:head.text=version
+0.866	-2:head.text=for	-0.912	-2:lemma_=spell
+0.820	-3:head.text=variation	-0.932	-3:head.text=misspelling
+0.789	-2:word.lower=word	-0.983	head_text=say
+0.776	-3:head.text=mispelling	-1.025	head_text=of
+0.724	-1:lemma_=word	-1.046	-3:lemma_=spelling
+0.658	-3:head.text=form	-1.055	-2:head.text=misspelling
+0.654	dep_=acomp	-1.110	head_text=saying
+0.647	-3:word.lower=word	-1.127	-3:lemma_=way
+0.594	-1:dep_=prep	-1.133	-3:word.lower=way
+0.586	-2:lemma_=misspelling	-1.148	head_text=for
… 165 more positive …		-1.160	-1:tag_=``
… 115 more negative …		-1.236	-1:word.lower=spelling
-0.718	lemma_=word	-1.474	-1:head.text=mispelling
-0.734	dep_=prep	-1.501	-1:word.lower=word
-1.060	dep_=det	-1.916	-2:lemma_=spelling

Extracted features for CRF

* 颜色深浅表示各个feature的contribution大小，绿色表示positive contribution，红色表示 negative contribution。

Results





Bootstrapping method
准确率 ~70%, 覆盖率(coverage)较低

Self-training based CRF method
准确率 >90%, 覆盖率(coverage)较高

JDAI Research

Evaluation

使用提取的spelling variants进行词向量评估

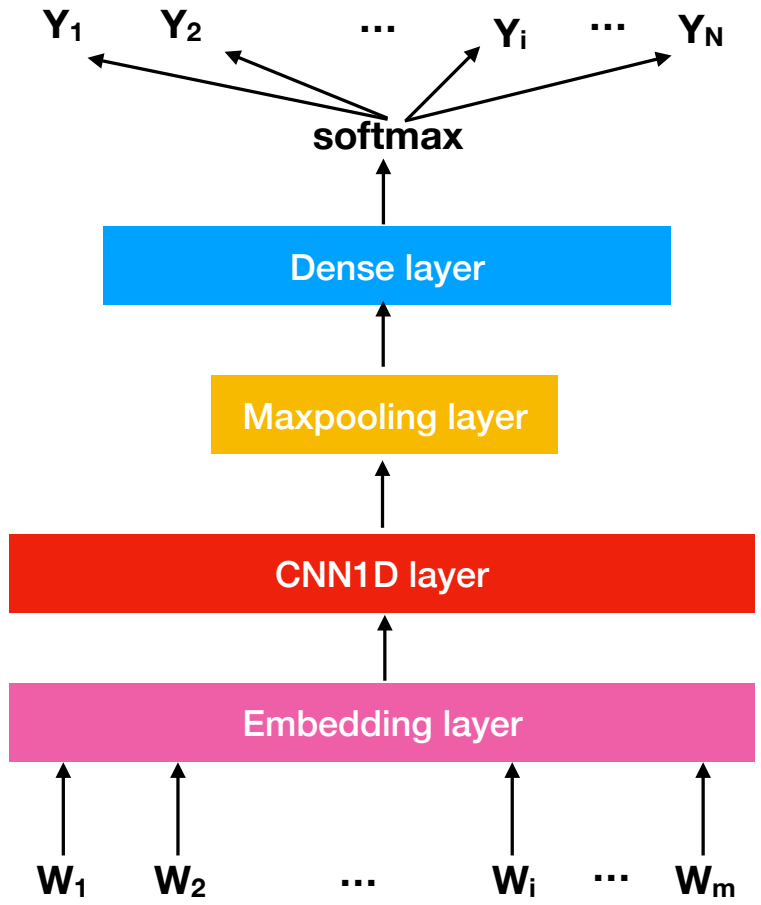
JD AI Research

评估预训练词向量

- 预处理语料：
  - 多语言推特：
    - 英文语言过滤，转换为小写，不进行text normalization （保留spelling variants）
- 预训练词向量
  - 训练语料: 3.5G Twitter corpora
  - 词向量训练方法：CBOW, skip-gram, GloVe, FastText，Mittens(adaptation)
- 词向量**内在评价**
  - 对于提取的variant spelling关系元组，计算对应预训练词向量的**MAP** (mean Average Precision)：如果当前variant spelling的词向量余弦相似度距离在当前词向量空间**排名**在**前k个** (此处k可看作超参数，取值为10，20，50，100)，认为precision取1，否则取0.
  - 利用MAP取值作为内在评价词向量的指标。

词向量外部评价

- 选取 Twitter 话题预测分类任务（ 此处为30分类）。
- 使用 textCNN 作为分类模型，模型在测试集上的准确率 > 90%；
- 词向量使用Word2Vec, fasttext, GloVe（同前面的内部评价）。



分类模型：textCNNs

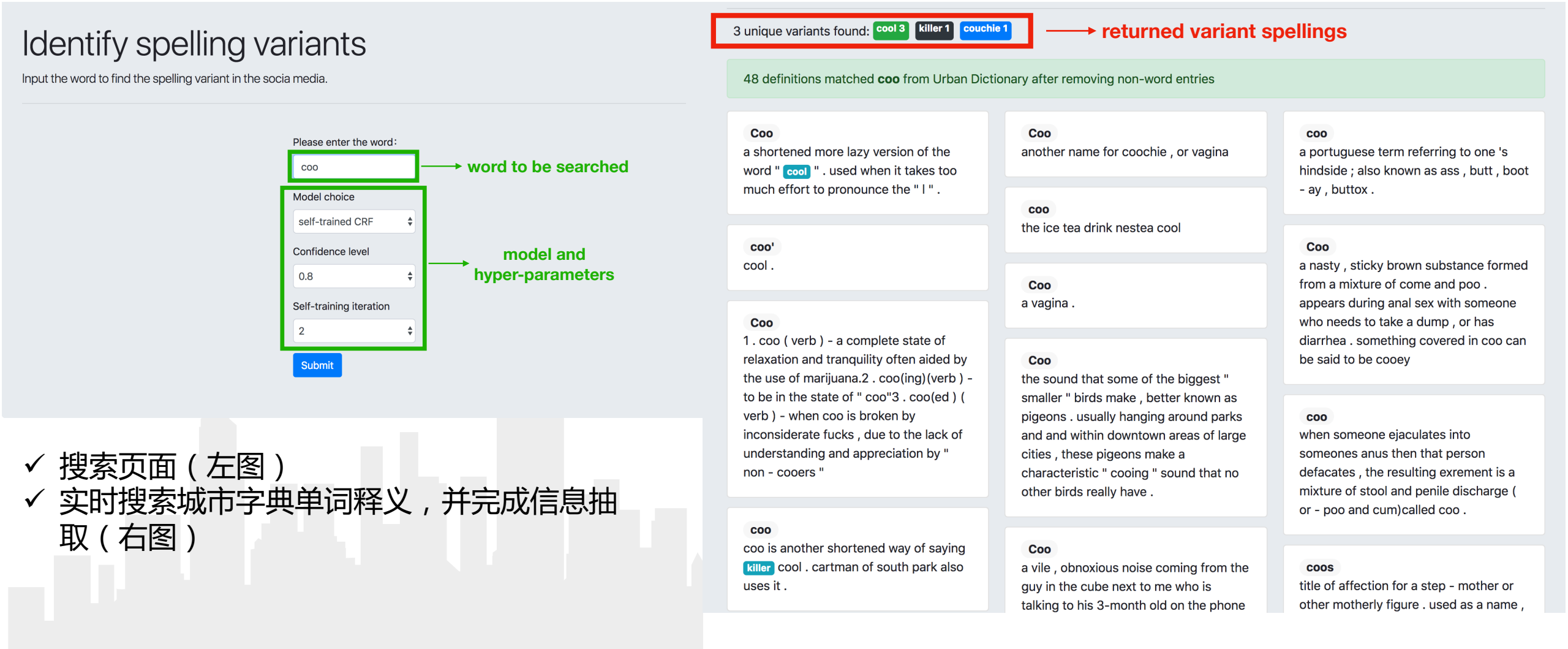
Comparison

内在评价 与 外在评价 进行相关性验证。

	val_acc	val_loss	acc	loss	top1-enwiki	top20-enwiki	top50-enwiki	top100-enwiki
val_acc	1.0	-0.81	0.45	-0.28	-0.52	-0.22	-0.014	-0.044
val_loss	-0.81	1.0	0.078	-0.28	0.51	0.5	0.3	0.18
acc	0.45	0.078	1.0	-0.97	-0.19	0.27	0.31	-0.0052
loss	-0.28	-0.28	-0.97	1.0	0.068	-0.43	-0.4	-0.093
top1-enwiki	-0.52	0.51	-0.19	0.068	1.0	0.62	0.68	0.7
top20-enwiki	-0.22	0.5	0.27	-0.43	0.62	1.0	0.84	0.73
top50-enwiki	-0.014	0.3	0.31	-0.4	0.68	0.84	1.0	0.8
top100-enwiki	-0.044	0.18	-0.0052	-0.093	0.7	0.73	0.8	1.0

Pearson correlation

在线英文 variant spelling实时搜索



- ✓ 搜索页面（左图）
- ✓ 实时搜索城市字典单词释义，并完成信息抽取（右图）

Contributions

- 提出一个大规模variant spelling数据集， 用于 informal domain 的NLP研究
- 利用非基于规则的半监督学习方法，对半结构化数据进行半监督提取，例如bootstrapping 方法。
- 基于条件随机场（CRF）使用自学习策略进行半监督序列标注，模型表现较好。
- 尝试使用提取的variant spelling数据集，通过衡量variant spelling的词向量的相似度距离排名，进行 Twitter 预训练词向量的内在评估。
- 实现在线variant spelling提取工具。

有什么**问题**呢？

- 多义词；
- 惯用语表示；
- 上下文独立；

(2018) 如何更好地表示词(们)？

NLP’s ImageNet moment has arrived !

July 2018

JDAI Research

NLP预训练的现代主义

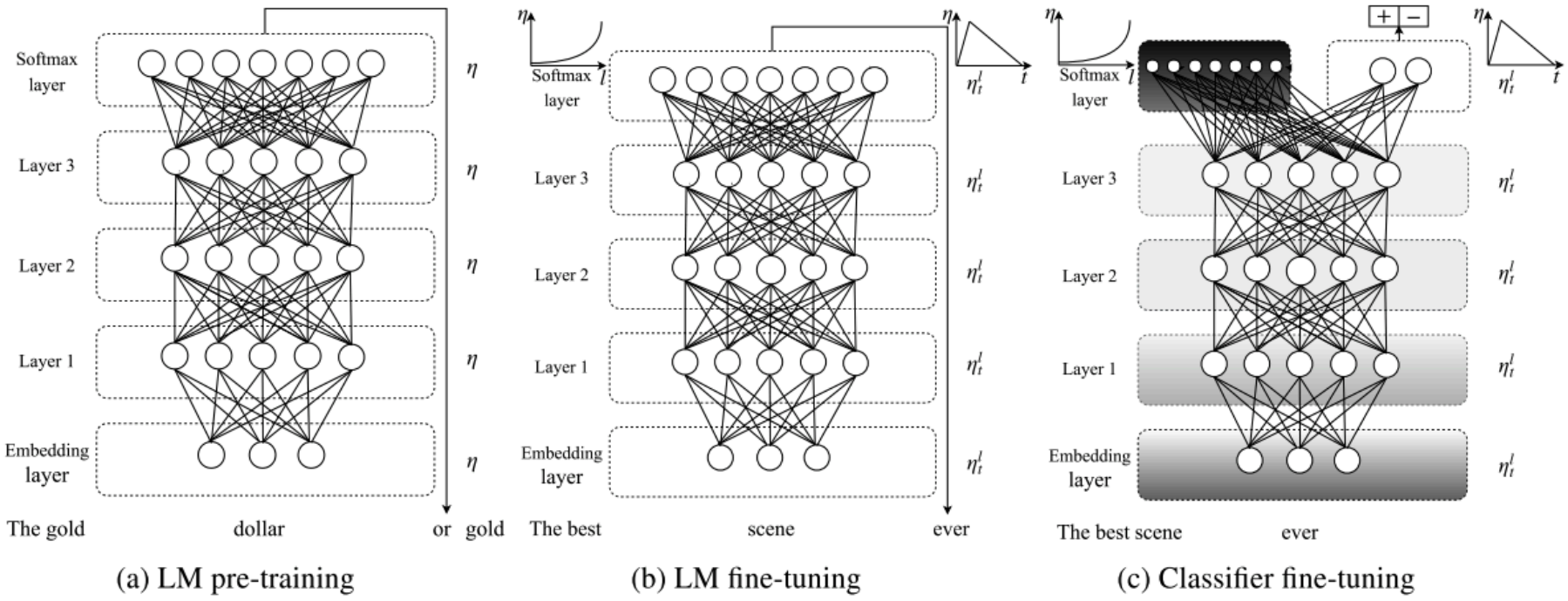
NLP2018

- ✓ UMLFit 路人甲
- ✓ tagLM -> ELMo 大表哥
- ✓ OpenAI Transformer GPT 大师兄
- ✓ BERT 二师兄 (公关做得好？)

ULMFit（通用语言模型微调）

- **问题**：使用分类器微调时，语言模型对小规模数据集过拟合 并且遭遇毁灭性遗忘

- 模型：AWD-LSTM
- 三步骤：
  - 通用domain语言模型训练
  - 目标任务语言模型微调
  - 目标任务分类器微调训练
- 解决的问题：
  - 消除毁灭性遗忘
  - 使迁移学习更加健壮(鲁棒)



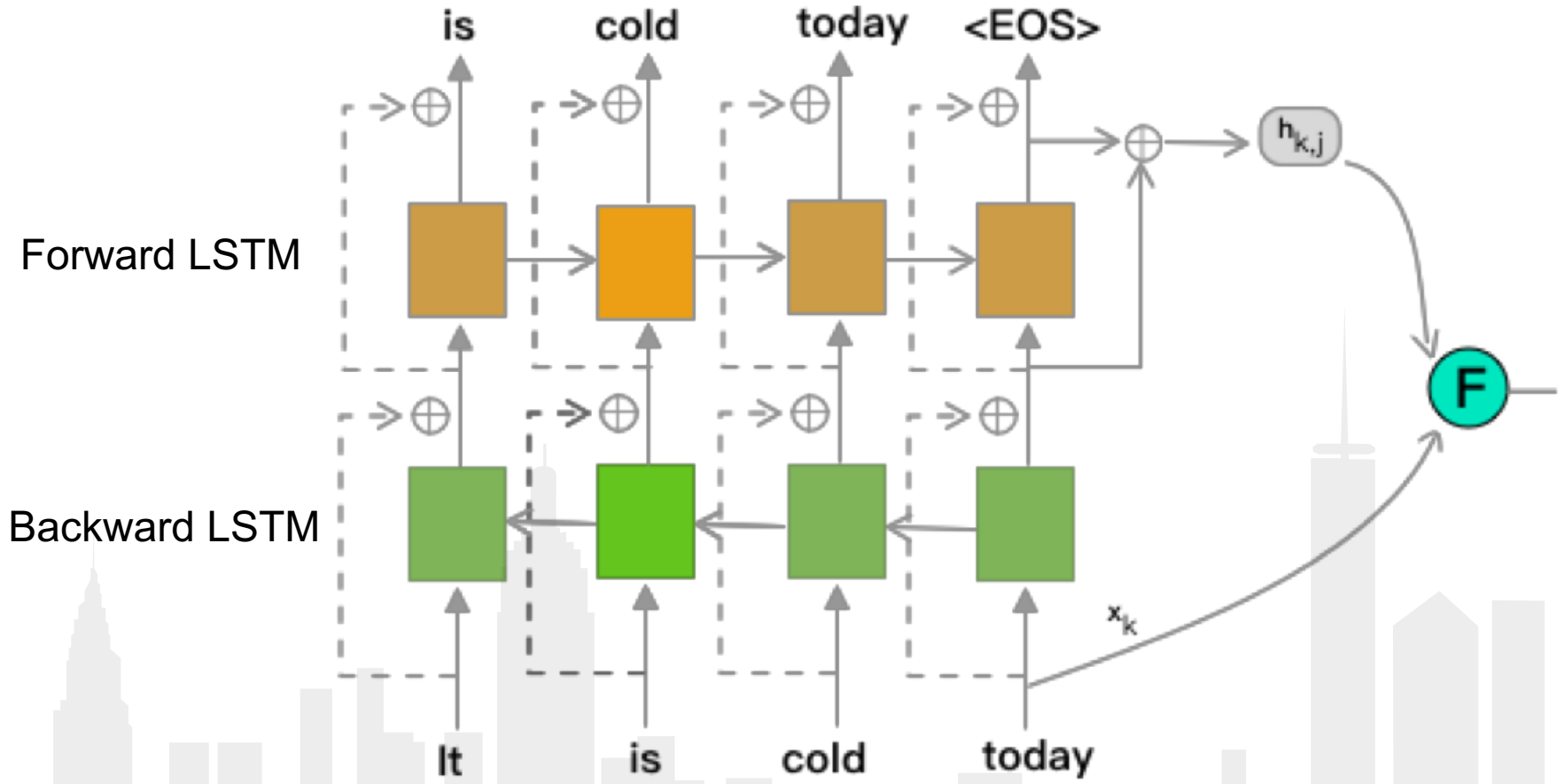
ELMo（NAACL 2018）

- **问题**：词表征上下文独立；只能对于句法和语义信息建模，而忽略不同语言环境下的多样性用法。
- 前人的改进：
  - 引入小于字级别的信息（fasttext, chargram）
  - 对不同的词义分别学习单独表示

- ELMo（来自语言模型的嵌入）
  - 提取上下文敏感特征对多义词进行建模
  - 深层次表示 → 双向语言模型所有内部各层状态信息的函数
  - 底层使用char ConvNets，引入字级别内部信息
  - 针对所有双向LSTM层的表示计算具体任务的权重

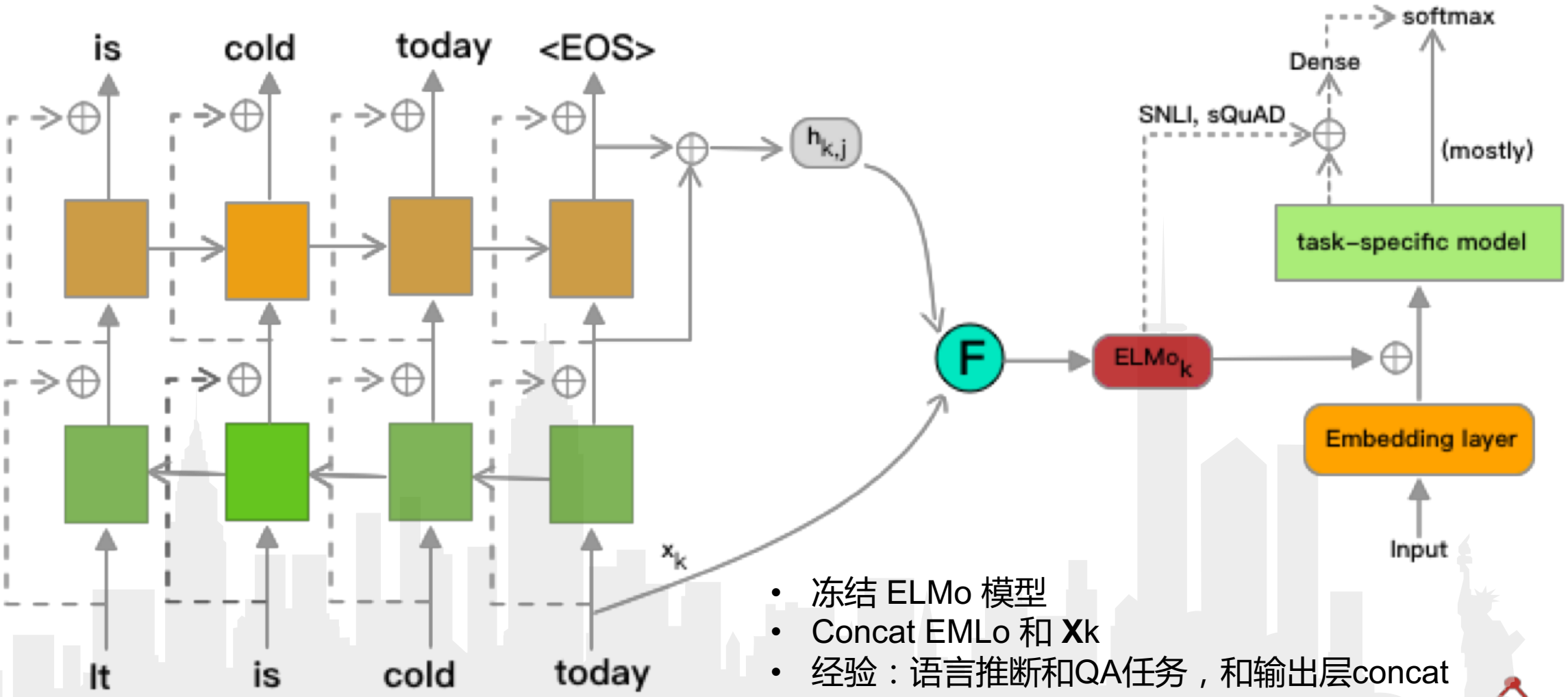
$$ELMo_{o_k}^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

ELMo 怎么训练？



高层LSTM状态捕捉上下文独立的词义信息（WSD）；低层信息捕捉到句法信息（POS tagging）
同时学习的所有层内部信息，在后续训练中被模型中以半监督方式F选择。

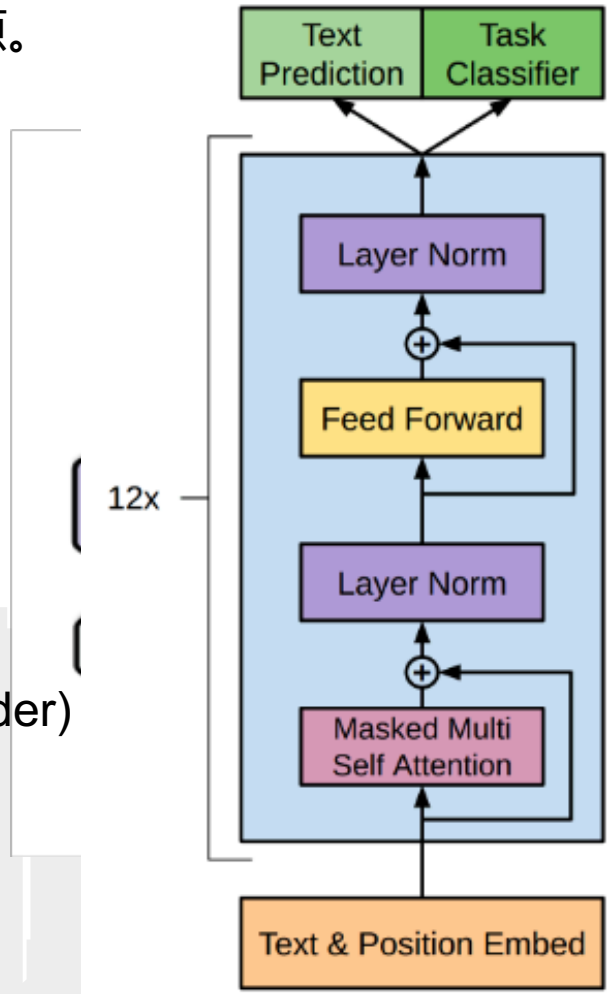
训练好了怎么使用？

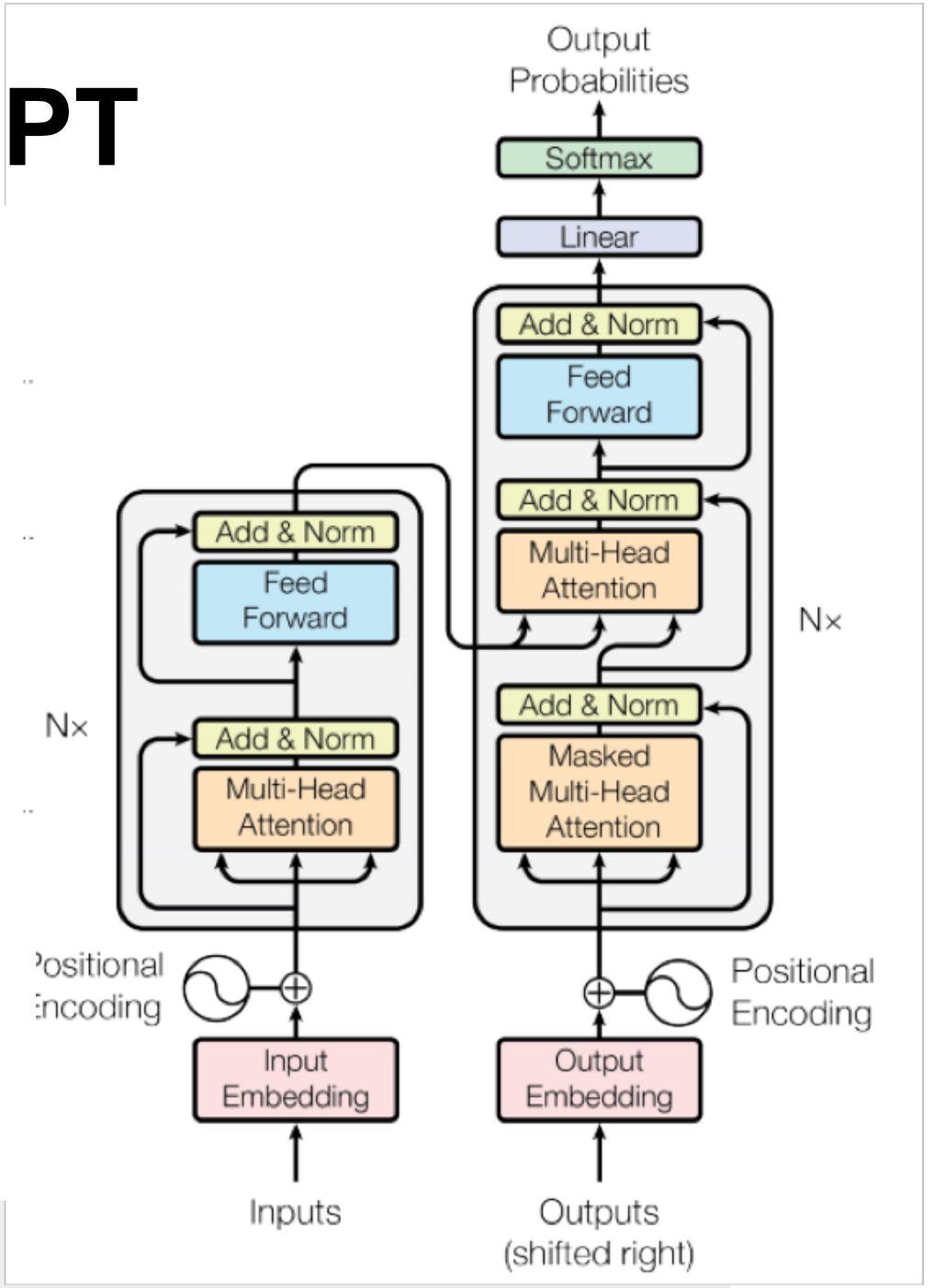


- 冻结 ELMo 模型
- Concat EMLo 和 **X**k
- 经验：语言推断和QA任务，和输出层concat

OpenAI Transformer GPT

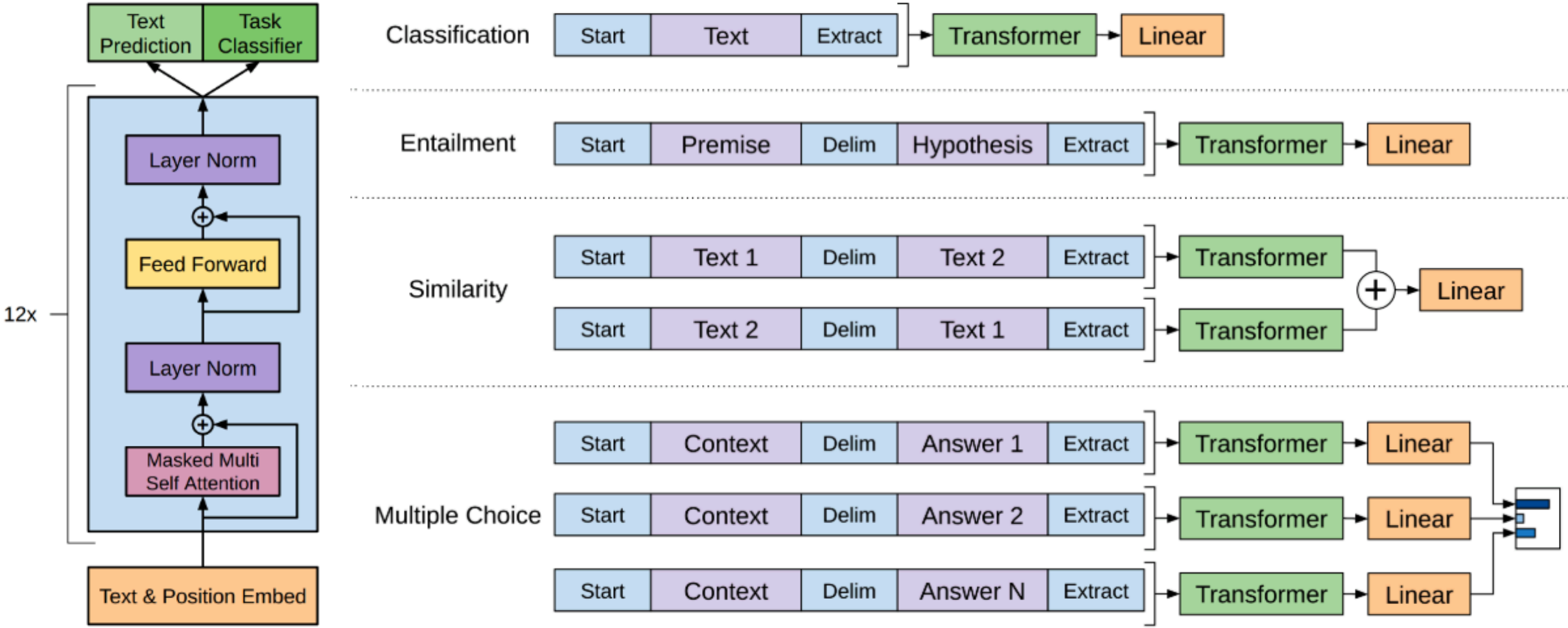
- **问题**：深度学习模型缺少大量标注资源。





- 模型：12 x forward Transformer (encoder)
- Use BPE
- 激活函数： Gelu
- 两步：
  - 语言模型训练
  - 具体人物模型训练
    - 改进：使用辅助目标函数，$+\lambda \mathrm{L}(LM)$

OpenAI Transformer GPT



BERT

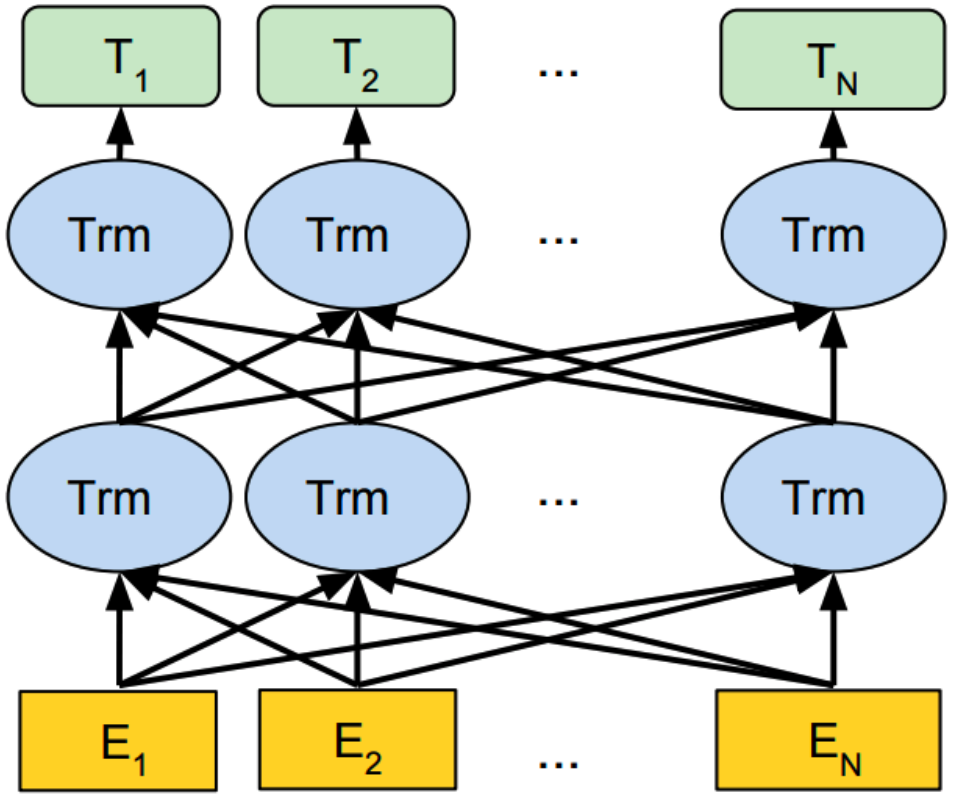
- 模型：多层双向Transformer
- 激活函数： GELU

语言模型训练

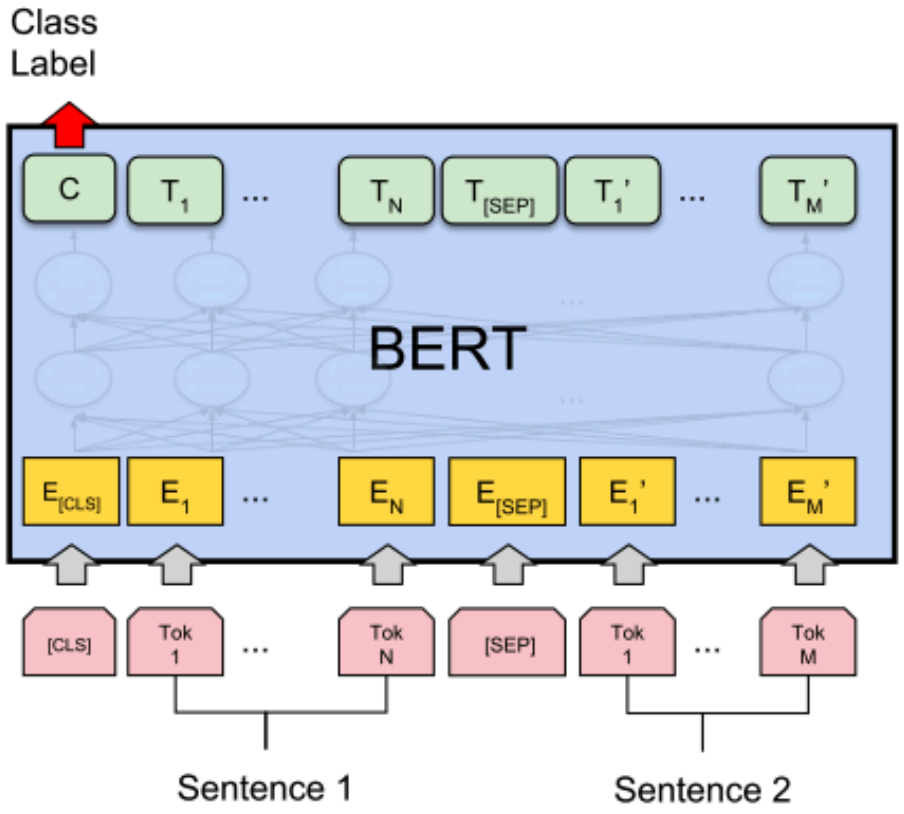
- **#1 Masked LM**
  - 随机mask 15%的词，预测mask的词
- **#2 Next Sentence Prediction**

MLM 带来的问题

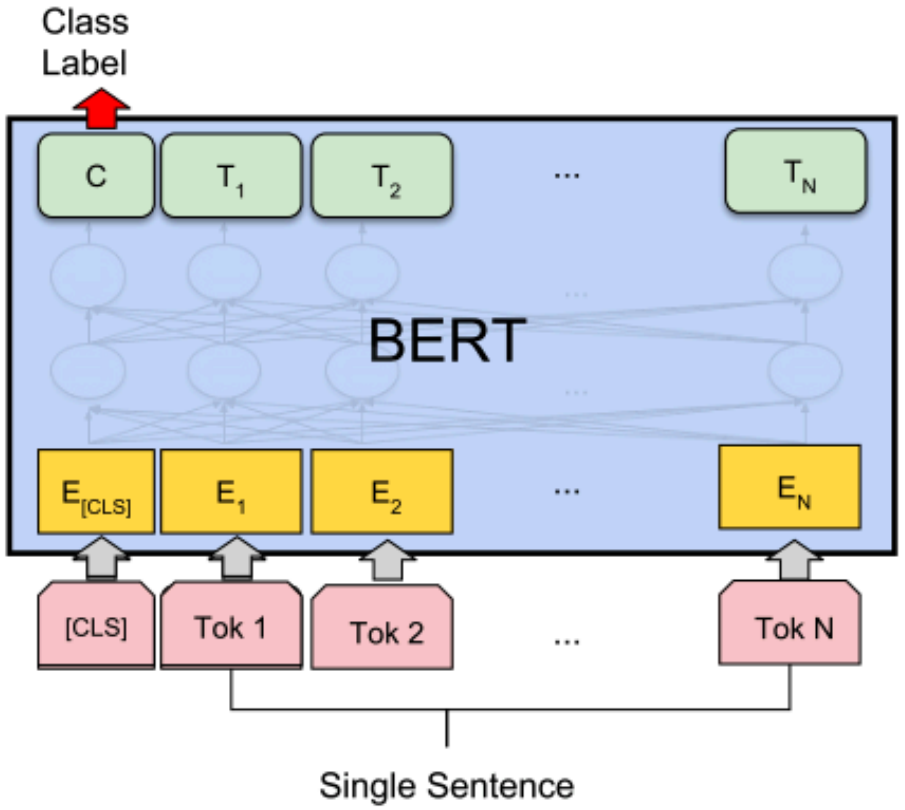
- **问题1：** Mask的词永远不可能被看到，预训练和微调过程不匹配
- 解决办法：80%情况替换成[MASK]，10%替换成随机的词，10%不替换
- 问题2: mask 15%的词导致拟合过程变长
- 解决办法：不解决。时间换效果。



BERT 怎么用？

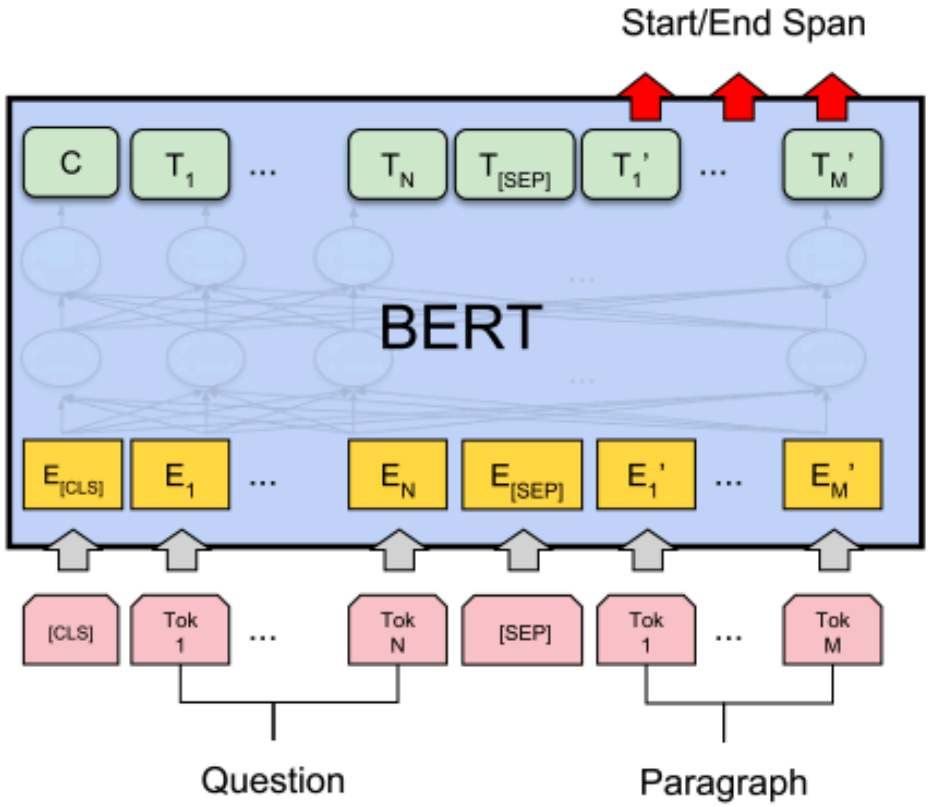


(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

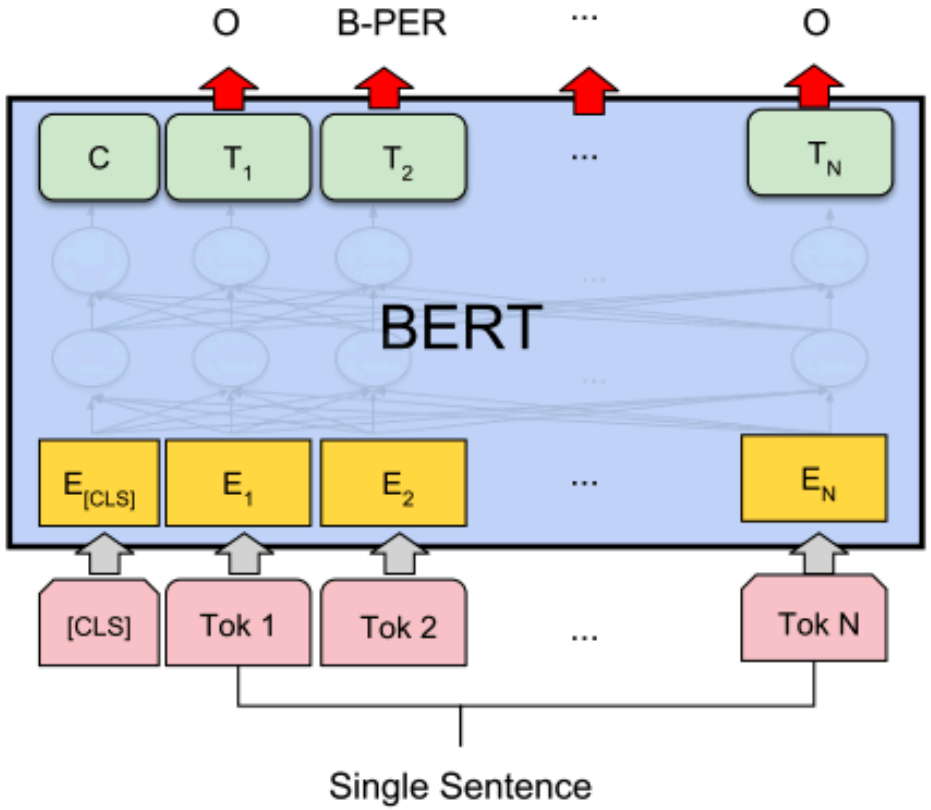


(b) Single Sentence Classification Tasks:
SST-2, CoLA

BERT 怎么用？



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Comparison between ELMo, OpenAI GPT and BERT

Model	architecture	pretraining task	Usage style
ELMo	bi-LSTM	LM	feature-based
OpenAI GPT	left-to-right Transformer	LM	fine-tuning
BERT	bi-Transformer	MLM, NSP	fine-tuning

Training time: Transformer < ConvNets < Simple RNNs < LSTMs .

要什么自行车？

- 自行车： RNN / LSTM / GRU
- 电动车： CNN
- 摩托车： **Transformer**
- 三轮摩托：？？
- 桑塔纳：？？
- 法拉利：？？？

期待NLP的后现代主义

“穷人“也能做NLP

JDAI Research

感谢大家的参与 ☺