自然语言处理中的文本例化

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

概要

- **文本例化概要**
- 基于词的文本例化
- 基于子词的文本例化

文本例化概要

- 自然语言是通过形式表达意义的系统
  - 是由有限基本单位(词汇)组成的符号串
  - 所有基本单位组成的集合叫作词汇表(vocabulary)
  - 文本是由词汇表中基本单位组成的符号串
  - 基本单位承载基本意义
  - 文本的意义可由基本单位的意义组合得到
    - 首先习得基本单位的意义
    - 在此基础上产生文本的意义

文本例化概要

- 文本例化
  - 把符号串形式的文本切分成基本组成单位的过程
  - 切分所得到基本单位被称作token
- 基本单位的选择
  - 词(word)
  - 子词(subword)
  - 字符(character)
- 文本例化策略
  - 基于词的文本例化
  - 基于子词的文本例化
  - 基于字符的文本例化

文本例化概要

- 基于词的文本例化
  - 将文本切分成词的序列 (字符串→词串)
    e.g. where is the tallest building
    ⇒ where/is/the/tallest/building
  - 词是可以独立运用的最小意义单位
  - 语言学驱动的策略，传统的文本例化策略
- 基于子词的文本例化
  - 将文本切分成子词序列(字符串→子词串)
    e.g. where is the tallest building
    ⇒ where/is/the/tall /est/ build/ ing
  - 目前使用较多的文本例化策略
- 文本例化是自然语言处理首先要做的事情

概要

- 文本例化概要
- **基于词的文本例化**
- 基于子词的文本例化

基于词的文本例化

- 基于词的文本例化方法 和 语种相关
- 英语等印欧语
  - 词和词之间有形式上的分隔符号
    - 空格、标点符号等
  - 可以基于这些分隔符号进行文本例化
- 汉语(日语等语言)
  - 按句连写，词和词之间没有分隔标记

    e.g. 我将出席会议 ⇒我/将/出席/会议
  - 文本例化成为有挑战性的任务

英语中的例化问题

- 英语也不能仅凭空格和标点符号解决切分问题
  1. 缩写词

     N.A.T.O. i.e. m.p.h Mr. AT&T
  2. 连写形式以及所有格词尾

     I’m He’d don’t Tom’s
  3. 数字、日期、编号

     128,236 +32.56 –40.23 02/02/94 02-02-94

     D-4 T-1-A B.1.2
  4. 带连字符的词

     text-to-speech text-based e-mail co-operate
- 与汉语文本例化相比，较为容易。(辅以规则)

汉语自动切分

- 基于词的汉语文本例化又叫汉语自动切分，是经典的汉语自然语言处理任务。(Chinese Word Segmentation)
- 基于词表的方法
  - 需要配备词表
  - 通过词表匹配确定字串是否成词
  - 规则驱动、数据驱动
- 字序列标记方法
  - 无需配备词表
  - 根据语境判断字在词中的位置
  - 数据驱动

基于词表的汉语切分

- 最为简单的词表法 最大匹配法
- 正向最大匹配法(MM)　　从左向右匹配词表
- 逆向最大匹配法(RMM)　　从右向左匹配词表
- 例子
  - 输入:　企业要真正具有用工的自主权
  - MM:　企业/要/真正/具有/用工/的/自主/权
  - RMM:　企业/要/真正/具有/用工/的/自/主权

最大匹配法的特点

- 算法简单
- 长词优先
  - 输入:　鱼在长江中游
  - MM:　鱼/在/长江/中游
  - RMM: 鱼/在/长江/中游

  - 长词优先是否合理？
  - 词表：中游、中、游 结果：　中游

字序列标记方法

- 词位标记
  (1) B 词首　　(2) M　词中
  (3) E 词尾　　(4) S　　单字成词
- 根据语境确定字在词中的位置标记
  她努力学习考上了北京大学
  她/S 努/B 力/E 学/B 习/E 考/S 上/S 了/S北/B 京/M 大/M 学/E
- 切分可看作给句中每个字加位置标记
- 设计给字序列加标记的模型和方法
- 有人称这种方法是“合”词法

汉语切分的关键问题

- **切分歧义消解**
  - **切分歧义**: 字串有存在多种切分方式

    鱼 在 长江 **中 游** ⇔ 鱼 在 长江 **中游**
- **未登录词识别**
  - **未登录词**(OOV): 词表中没有收录或者训练语料库中没有出现的词
  - 人名、地名、科技术语、新词

- 实践中，未登录词造成的影响更加严重

切分歧义类型

1. 交集型歧义
   - 字串AJB中，若AJ、JB、A、B都是词，则AJB会有AJ/B、A/JB两种切分方式。称字串AJB是交集型歧义字段，其中J为交集字段

     从小学

     从小/**学**/电脑 **从**/**小学**/毕业
2. 组合型歧义
   - 字串AB中，若AB、 A、 B都是词，则AB会有AB、A/B两种切分方式。称字串AB是组合型歧义字段。

     中将

     美军/**中将**/竟公然说 新建地铁/**中**/**将**/禁止商业摊点

切分歧义类型

- 交集型歧义的**链长**
  - 交集型歧义字段中交集字段的个数，称作链长

从小学	链长是1
结合成分	链长是2
为人民工作	链长是3
中国产品质量	链长是4
部分居民生活水平	链长是6
治理解放大道路面积水	链长是8

真歧义与伪歧义

1. 真歧义
   - 歧义字段在不同语境中确有多种切分方式

     地面积

     这块/**地**/**面积**/还真不小

     地面/**积**/了厚厚的雪

     和平等

     自由/**和**/**平等**/是否具有内在矛盾性

     阿美首脑会议将讨论巴以/**和平**/**等**/问题

     把手

     卧室门/**把手**/坏了

     别/**把**/**手**/伸进别人的口袋里

真歧义与伪歧义

2. 伪歧义
   - 歧义字段单独拿出来看有歧义，但在(所有)真实语境中只有一种切分方式可接受

     挨批评

     挨/批评(√)　**挨批/评**(×)

     学生/**挨**/**批评**/挥拳打老师

     平淡

     平淡(√)　**平**/**淡**(×)

     平淡/生活感动人

歧义的发现

- 歧义消解的前提是发现歧义
  - 切分算法应有发现输入文本中是否出现歧义切分字段的能力
- MM和RMM均没有检测歧义的能力
  - 只能给出一种切分结果

歧义的发现

- 双向最大匹配(MM+RMM)
  - 同时使用MM法和RMM法
  - 如果MM法和RMM法给出同样的结果，认为没有歧义，若不同，则认为出现了歧义

    输入：企业要真正具有用工的自主权
    MM：企业/要/真正/具有/用工/的/自主/权
    RMM: 企业/要/真正/具有/用工/的/自/主权

歧义的发现

- 双向最大匹配法不能发现所有的歧义，有盲点
  - 不能发现组合型歧义 (长词优先)

    输入:　他从**马上**下来
    MM:　他/从/**马上**/下来
    RMM:　他/从/**马上**/下来
    正确:　他/从/**马**/**上**/下来
  - 链长是偶数时，不能发现交集型歧义

    输入:　原子**结合成分子时**
    MM:　原子/**结合**/**成分**/**子时**
    RMM:　原子/**结合**/**成分**/**子时**
    正确:　原子/**结合**/**成**/**分子**/**时**

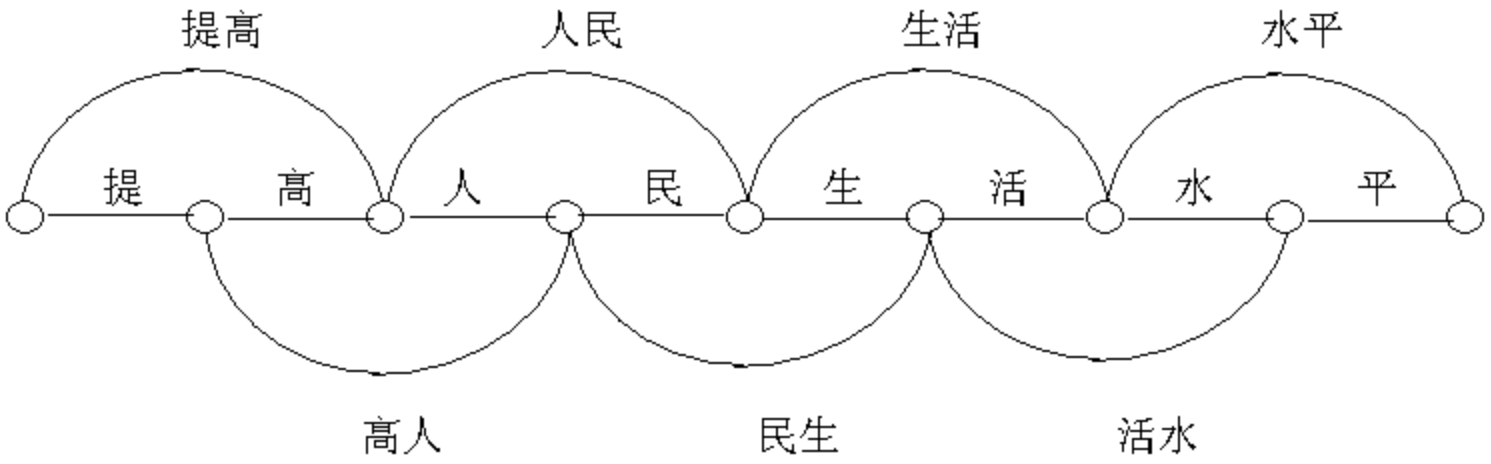
歧义的发现

- 发现组合型歧义
  - MM+逆向最小匹配法
- 发现所有切分歧义
  - 全切分算法

输入: 提高人民生活水平
输出: 提/高/人/民/生/活/水/平
提高/人/民/生/活/水/平
提高/人民/生/活/水/平
提高/人民/生活/水/平
提高/人民/生活/水平
……

数据结构

- 歧义切分的表示—词图



歧义消解

- 基于记忆的伪歧义消解
  - 伪歧义所占比例非常大
  - 伪歧义消解与上下文无关，对高频伪歧义字段，可把它们的正确(唯一)切分形式预先记录在一张表中，其歧义消解通过直接查表即可实现。

歧义消解

- 基于规则的歧义消解

  规则1：P[+R+M+Q+A|Z]+”马上”→马+上
  他**从**大红/马/上/下来
  这件事需要/马上/办

  规则2：“一起”+~V→一+起
  我们/一起/去故宫
  一/起/恶性交通事故

- 注意规则中字母的含义

歧义消解

- 基于统计的歧义消解
  - 在词图上搜寻统计意义上的最佳路径

$$\hat{p} = \underset{p_i \in G}{\operatorname{argmax}}\, score(p_i)$$

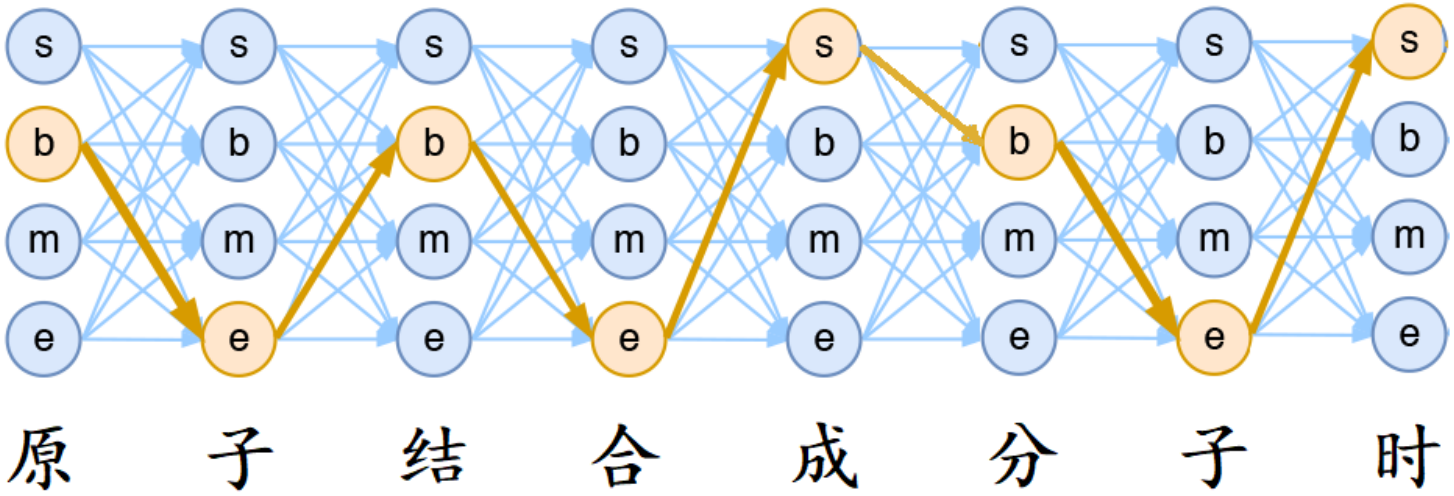
- 定义路径分值
  - 基于n元模型，计算路径概率
  - …

$score(\text{原子/结合/成分/子时})$
$= p(\text{原子})p(\text{结合}|\text{原子})p(\text{成分}|\text{结合})p(\text{子时}|\text{成分})$
$score(\text{原子/结合/成/分子/时})$
$= p(\text{原子})p(\text{结合}|\text{原子})p(\text{成}|\text{结合})p(\text{分子}|\text{成})p(\text{时}|\text{分子})$

字序列标记法

- 求解最佳标记序列

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{x})$$



字序列标记法

- 条件随机场模型

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{c \in C} \sum_{i} \lambda_i f_i(\boldsymbol{y_c}, \boldsymbol{x})\right)$$

- 特征选择
  - $C_n \ (n = -2, -1, 0, 1, 2)$
  - $C_n C_{n+1} \ (n = -2, -1, 0, 1)$
  - $C_{-1} C_1$

$$f_i = \begin{cases} 1 & C_0 = \text{子} \ \& \ y_0 = E \\ 0 & otherwise \end{cases}$$

$$f_j = \begin{cases} 1 & C_0 = \text{子} \ \& \ y_{-1} y_0 = BE \\ 0 & otherwise \end{cases}$$

未登录词识别

- 中国人名：李素丽 老张 李四 王二麻子
- 中国地名：定福庄 白沟 三义庙 韩村河 马甸
- 翻译人名：乔治·布什 叶利钦 包法利夫人 酒井法子
- 翻译地名：阿尔卑斯山 新奥尔良 约克郡
- 机构名：方正公司 联想集团 国际卫生组织 外贸部
- 商标字号：非常可乐 乐凯 波导 杉杉 同仁堂
- 专业术语：万维网 主机板 模态逻辑 贝叶斯算法
- 缩略语：三个代表 五讲四美 打假 扫黄打非 计生办
- 新词语：温拿、卢瑟、给力、吊丝、骚年

未登录词识别

- 未登录词识别困难
  - 缺乏词表或标注语料指导信号
  - 边界不易确定
- 传统上，逐类构造专门未登录词识别算法
- 在字序列标注法中，通常不做未登录词识别
- 识别依据
  - 内部构成规律（用字规律）
  - 外部环境（上下文）

未登录词识别

- 未登录词识别的的难度
  - 较容易
    - 中国人名、译名
    - 中国地名
  - 较困难
    - 商标字号
    - 机构名
  - 很困难
    - 专业术语
    - 缩略语
    - 新词语

中文人名识别

- 在汉语的未登录词中，中国人名是规律性最强，也是最容易识别的一类；
  - 中国人名一般由以下部分组合而成：
    - 姓：张、王、李、刘、诸葛、西门
    - 名：李素丽，王杰、诸葛亮
    - 前缀：老王，小李
    - 后缀：王老，赵总
  - 中国人名各组成部分用字比较有规律

中文人名识别

- 根据统计, 汉语姓氏大约有1000多个(**数量有限**), 姓氏中使用频度最高的是“王”姓, “王, 李, 张, 刘, 陈”等5个大姓覆盖率达32%, 姓氏频度表中的前14个高频度的姓氏覆盖率为50%, 前400个姓氏覆盖率达99%。
- 人名的用字也比较集中。频度最高的前6个字覆盖率达10.35%, 前10个字的覆盖率达14.936%, 前15个字的覆盖率达19.695%, 前400个字的覆盖率达90%

自动切分评价指标

- **准确率**(P)

$$P = \frac{\text{切分结果中正确分词数}}{\text{切分结果中所有分词数}} \times 100\%$$

- **召回率**(R)

$$R = \frac{\text{切分结果中正确分词数}}{\text{标准答案中所有分词数}} \times 100\%$$

- **F-值**(综合指标，P和R的调和平均值)

$$F\text{−值} = \frac{2PR}{P+R}$$

- 以词作为评价单位

汉语自动分词的评测

- 国际 ACL SIGHAN bakeoff (2003~2007)
  后改为和中文信息学会联合举办

Site	word count	R	c_r	P	c_p	F	OOV	R_{oov}	R_{iv}
S01	17,194	0.962	±0.0029	0.940	±0.0036	0.951	0.069	0.724	0.979
S10	17,194	0.955	±0.0032	0.938	±0.0037	0.947	0.069	0.680	0.976
S09	17,194	0.955	±0.0032	0.938	±0.0037	0.946	0.069	0.647	0.977
S07	17,194	0.936	±0.0037	0.945	±0.0035	0.940	0.069	0.763	0.949
S04	17,194	0.936	±0.0037	0.942	±0.0036	0.939	0.069	0.675	0.955
S08	17,194	0.939	±0.0037	0.934	±0.0038	0.936	0.069	0.642	0.961
S06	17,194	0.933	±0.0038	0.916	±0.0042	0.924	0.069	0.357	0.975
S05	17,194	0.923	±0.0041	0.867	±0.0052	0.894	0.069	0.159	0.980

- 封闭 /开放 （是否可以使用训练语料之外的其它语言资源）
- 多个训练语料，回避标准问题

什么是词？

- 词是由语素构成的、能够独立运用的最小的语言单位。
- 缺乏操作标准。
- 汉语中语素、词和词组的界线模糊。
  - 象牙 是词？ 兔牙？
  - 吃饭 吃鱼
  - 毁坏 打坏

什么是词？

- 关于什么是词，不同的人有不同的把握[1]。

	M1	M2	M3	T1	T2	T3
M1		0.77	0.69	0.71	0.69	0.70
M2			0.72	0.73	0.71	0.70
M3				0.89	0.87	0.80
T1					0.88	0.82
T2						0.78

100个句子（4372字），6个人 人工切分，两两比较

Sproat R. et al. 1996. A Stochastic Finite-state Word Segmentation Algorithm for Chinese. Computational Linguistics, Vol.22 No.3, P377-404.

汉语分词规范

- 《信息处理用汉语分词规范》GB/T13715-92，中国标准出版社，1993
  - **分词单位**：汉语信息处理使用的、具有确定的语义或语法功能的基本单位。包括本规范的规则限定的词和词组。
  - 规范按词类分别给出了各类分词单位的定义，并给出例子。
  - 规范中多处使用了“**结合紧密、使用稳定**”的表述
  - 不但有规范 还要有词表（还要有语料）
- 《资讯处理用中文分词规范》台湾中研院，1995
- 切分单位的确定和应用有关

概要

- 文本例化概要
- 基于词的文本例化
- **基于子词的文本例化**

未登录词问题

- 训练语料或词表不可能收全所有的词
- 无论中文、英文都有未登录词问题(unknown word)
- 词的界定存在困难

- 受限于训练语料或词典
  - 模型只能习得已登录词的意义表示
  - 测试语料中的未登录词无法获得有效处理

- 工程上的解决办法，引入特殊词例<UNK>
  - 将训练数据中的低频词替换为<UNK>，习得<UNK>的意义表示
  - 将所有未登录词统一处理为<UNK>
  - <UNK>可能代表很多不同的词

基于子词的文本例化

- 把文本切分为比词小的单位——子词
- 收集所有子词并习得其意义表示
- 词的意义可以由子词的意义加以推断
- 词是由语素构成的

  e.g. unhappiness → un-happy-ness
  - 习得语素的意义推断词的意义
  - 语言学支持
- 基于语素的例化并非好的选择
  - 需要词根级别的形态分析器(morphological parser)
  - 依然存在未登录词(语素)

基于子词的文本例化

- 基于训练语料自动提取子词词表(subword vocabulary)

- 子词可能是任意字符组合
  - 不要求具有语言学上的意义，比如对应语素
  - 子词长度灵活
    - 可以长(完整的词)、也可以短(甚至是一个字符)
  - 常用词例化为词，利于准确习得意义
  - 生僻词分解为子词，利于推断未登录词的意义
  - 子词词表的大小可以灵活控制

BPE子词例化

- BPE例化算法原是一种无损数据压缩方法，2016年被Sennrich用来进行文本例化
  - 生成子词词表
  - 对文本例化
- 生成子词词表的过程
  - 对训练语料进行预切分(pre-tokenization)
    - 利用空格等信息、利用规则等
    - 利用基于词的例化方法
  - 基于预切分结果，生成频率词典
  - 基于BPE算法生成子词词表——不断合并高频二元子词串

- 假定训练语料是： Pen Penapple Apple Pen
- 生成频率词典

  2: pen

  1: penapple

  1: apple
- 生成初始子词表(每个字符作为一个子词)

  2: p e n _

  1: p e n a p p l e _

  1: a p p l e _

  V=[_, p, e, n, a, l]
- **统计**高频二元子词串 {(p, e): 3, (e, n): 3, (n, _): 2, (a, p): 2, (p, p): 2, (p, l): 2, (l, e): 2, (e, _): 2, (n, a): 1}
- 选频率最高的二元子词串**合并**，加入子词表 V=[_, p, e, n, a, l, pe]
- 更新频率词典

  2: pe n _

  1: pe n a p p l e _

  1: a p p l e _

2: pe n _

1: pe n a p p l e _

1: a p p l e _

- 统计 {(pe, n): 3, (n, _): 2, (a, p): 2, (p, p): 2, (p, l): 2, (l, e): 2, (e, _): 2, (n, a): 1}
- 合并 V= [_, p, e, n, a, l, pe, pen]
- 更新

  2: pen _

  1: pen a p p l e _

  1: a p p l e _

- 统计 {(pen, _): 2, (a, p): 2, (p, p): 2, (p, l): 2, (l, e): 2, (e, _): 2, (pen, a): 1}
- 合并 V= [_, p, e, n, a, l, pe, pen, pen_]
- 更新

  2: pen_

  1: pen a p p l e _

  1: a p p l e _

合并　　当前子词表

(a, p)　　[_, p, e, n, a, l, pe, pen, pen_, ap]

(ap, p)　　[_, p, e, n, a, l, pe, pen, pen_, ap, app]

(app, l)　　[_, p, e, n, a, l, pe, pen, pen_, ap, app, appl]

(appl, e)　　[_, p, e, n, a, l, pe, pen, pen_, ap, app, appl, apple]

(apple, _)　　[_, p, e, n, a, l, pe, pen, pen_, ap, app, appl, apple, apple_]

(pen, apple_)　[_, p, e, n, a, l, pe, pen, pen_, ap, app, appl, apple, apple_, penapple_]

- 假定待例化的文本是 Applepen PenapplePen
- 初始化(每个字符作为子词) a p p l e p e n _ p e n a p p l e p e n _
- 按照习得的顺序应用合并操作

  (p, e), (pe, n), (pen, _), (a, p), (ap, p), (app, l), (appl, e), (apple, _), (pen, apple_)

第1次合并 a p p l e pe n _　　pe n a p p l e pe n _

第2次合并 a p p l e pen _　　pen a p p l e pen _

第3次合并 a p p l e pen_　　pen a p p l e pen_

……

第9次合并 apple pen_　　pen apple pen_

BPE子词例化

Algorithm: Byte-pair encoding

Input: set of strings D, number of merges k

procedure BPE(D, k)

$V \leftarrow$ all unique characters in D

for $i = 1$ to k do

$t_L, t_R \leftarrow$ Most frequent bigram in D

$t_{NEW} \leftarrow t_L + t_R$

$V \leftarrow V + t_{NEW}$

Replace each occurrence of t_L, t_R in D with t_{NEW}

end for

return V

end procedure

BPE子词例化

- 在算法中，k用来控制子词词表的大小
- 子词的切分不跨越词的边界，需要做词的预切分
- 常用词例化为词，生僻词例化为子词
- 对于给定文本，可以
  - 顺序执行所习得的合并操作
  - 基于子词词表使用正向最大匹配法
- 确定性切分，只有一种切分结果
- GPT、RoBERTa、XLM等预训练模型采用BPE子词例化策略

WordPiece子词例化

- WordPiece也是一种常见的子词例化方法，其思想最早由Google公司的Schuster于2012年提出。

- WordPiece算法的核心也是构造子词词表，过程与BPE算法类似，从一个初始的子词词表(由字符组成)开始，不断合并二元子词串加入词表，达到创建子词词表的目的。

- 在得到子词词表后，也可以通过顺序执行合并操作或者最大匹配法完成文本例化

WordPiece子词例化

- BPE是选择频率最高的二元子词串合并。WordPiece采用最大似然原则选择合并的子串。
- 基于unigram计算训练语料的似然值(likelihood)。
- 选择能最大提升似然值的二元子词串合并
- 合并($t_x\ t_y \Rightarrow t_z$)所引起的对数似然值变化

$$\log P(t_z) - \left(\log P(t_x) + \log P\left(t_y\right)\right) = \log\left(\frac{P(t_z)}{P(t_x)P\left(t_y\right)}\right)$$

- 与BPE相比，运算复杂度较高
- BERT、DistilBERT、Electra等预训练模型采用WordPiece子词例化策略

Unigram LM子词例化

- BPE、WordPiece子词例化结果是确定性的
- 给定子词词表，对给定文本实际存在多种切分可能
- 下游模型中，可能需要概率信息，或需要多种切分结果
- 令X为待切分句子，$\mathbf{x} = (x_1, x_2, \cdots, x_M)$为一种子词切分形式，则基于unigram模型

$$P(\mathbf{x}) = \prod_{i=1}^{M} p(x_i)$$

- 令$S(X)$代表句子X所有切分形式，可以计算概率最大的切分形式

$$\mathbf{x}^* = \underset{\mathbf{x} \in S(X)}{\operatorname{argmax}} P(\mathbf{x})$$

Unigram LM子词例化

- 给定子词词表，如何计算子词概率$p(x_i)$?
  EM算法，最大化似然概率

$$L = \sum_{s=1}^{|D|} \log(P(X^{(s)})) = \sum_{s=1}^{|D|} \log\left(\sum_{\mathbf{x} \in S(X^{(s)})} P(\mathbf{x}) \right)$$

- Unigram LM子词例化基于Unigram模型确定子词词表并给出切分概率
- 与BPE、WordPiece不同，Unigram采用删除法构建词表，需要预先运用启发式策略产生一个大型种子词表，在此基础上逐步删除，直到子词词表规模满足要求
  字符表 + 高频子串

Unigram LM子词例化

- 子词删除原则
  - 删除后导致训练语料似然值损失($loss$)较小
  - 效果上会删除低频子词
- 基本过程

(1) 基于训练语料，运用启发式策略，建立种子词表V
(2) 重复下面的过程，直至词表规模$|V|$符合预先设定的大小
  (a) 固定词表，利用EM算法优化$p(x)$.
  (b) 对词表中子词x_i计算$loss_i$。
  (c) 根据$loss_i$对词表中的子词排序，保留排在前$\eta\%$的子词，更新词表V

- 复杂度较高，可以支持需要概率信息或多种结果的下游场景
- 也有预训练模型采用Unigram子词例化(AlBERT、T5、XLNet等)

SentencePiece

- BPE、wordPiece等子词切分无法做到语种无关
  - 需要基于词的预切分
  - 预切分因语言不同而异
  - 目的是子词不跨越词的边界

- SentencePiece目标是语种无关的子词切分
  - 不需要基于词的预切分、独立于语种
  - 直接基于训练文本构造子词词表
  - 空白符号与其他符号等同处理(子词中可能会有空白符号)
  - 是软件实现，不是具体的子词例化方法
  - 包含对BPE、Unigram LM和WordPiece的改进实现