



北京航空航天大學
BEIHANG UNIVERSITY

自然語言處理

人工智能研究院

主讲教师 沙磊



现代汉语词 语切分研究

什么是汉语自动切分？

- 通过计算机把组成汉语文本的字符串自动转换为词串的过程被称为自动切分(segmentation)。
 - 例子：
 - ◆ 鱼在长江中游
 - ◆ →鱼/在/长江/中/游
- 汉语和英语等印欧语不同，词和词之间没有空格。
 - 例子：
 - ◆ I'm going to show up at the Conference.

英语中的切分问题

- 英语中不是完全没有切分问题，不能仅仅凭借空格和标点符号解决切分问题。
 - 缩写词如：
 - N.A.T.O i.e. m.p.h Mr. AT&T
 - 连写形式以及所有格词尾
 - I'm He'd don't Tom's
 - 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
 - 带连字符的词
 - text-to-speech text-based e-mail co-operate

英语中的切分问题

- 英语中的切分通常被叫做Tokenization。
- 同汉语相比，英语切分问题较为容易。

为什么要进行汉语的切分研究

- 对汉语进行切分是许多应用的要求
- 1.TTS或语音合成
 - 只有正确切词，才能知道正确的发音，如：的(de0) 目的(di4)
 - 只有正确切词，才能正确变音，如：

(Third Tone Sandhi) 3+3→2+3 很好好酒 小老鼠 3+3+3 →
2+3+3 or 3+2+3

- 只有正确切词，才能正确解决轻声的问题，如：
冬瓜 桌子

为什么要进行汉语的切分研究

2. 信息检索

- 切分有助于提高信息检索的准确率，如：
- a. 和服务于三日后裁制完毕，并呈送将军府中。
- b. 王府饭店的设施和服务是一流的。

3. 词语的计量分析

- 词频统计(汉语中最常用的词是哪个词？)

4. ...

- 汉语切词也是深层汉语分析的基础
- 句法分析、语义分析等

基本方法

◆ 基于词表的方法

- 最大匹配法(MM)
 1. 正向最大匹配法(MM)
 2. 逆向最大匹配法(RMM)
 - 3. ...
- 全切分+路径选择

◆ 字序列标记方法

正向最大匹配法

```
S←待切分的字串;  
Segmentation←"";  
len ← maxlen;  
WHILE S≠"" DO  
    W ← substr(S,0,len);  
    IF (W∈D) THEN /*D 为电子词典*/  
        S ← S - W;  
        Segmentation ← Segmentation + W + "/";  
        len ← maxlen;  
    ELSE  
        IF len = 1 THEN  
            S ← S - W;  
            Segmentation ← Segmentation + W + "/";  
            len ← maxlen;  
        ELSE  
            len ← len - 1;  
        ENDIF  
    ENDIF  
END WHILE
```

逆向最大匹配法

- 正向最大匹配法从左向右匹配词典
- 逆向最大匹配法从右向左匹配词典
- 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

- 长词优先
 - 输入:他将来中国
 - MM:他/将来/中国
 - RMM:他/将来/中国
 - 正确:他/将/来/中国
- 算法非常简单

序列标注方法

- 把切分问题看作给句子中每个字加标记的过程。 四个标记：
- (1) B 词首 (2) M 词中
- (3) E 词尾 (4)单独成词S
- 例如：
- 输入:提高人民的生活水平
 - 提/B 高/E 人/B 民/E 的/S 生/B 活/E 水/B 平/E
- 设计一个给字序列标注标记序列的算法

自动切分的评价

- 准确率(precision)

- 准确率 (P) = 切分结果中正确分词数 / 切分结果中 所有分词数 * 100%

- 召回率(recall)

- 召回率 (R) = 切分结果中正确分词数 / 标准答案中 所有分词数 * 100%

- F-评价(F-measure 综合准确率和召回率的评价指标)

- F-指标 = $2PR/(P+R)$

汉语切分的关键问题

- 切分歧义（消解）
 - 一个字串有不止一种切分结果
- 未登录词识别
 - 专有名词
 - 新词
- 据文献[1]，未登录词造成的影响更加严重
- “在大规模真实文本中未登录词造成的分词精度失落
- 比歧义切分造成的精度失落至少大5倍以上”

[1]黄昌宁、赵海，2007，中文分词十年回顾。《中文信息学报》第3期，8-19页。

切分歧义

1. 交集型歧义

- 字串 **AJB** 中，若 $AJ \in D$ 、 $JB \in D$ 、 $A \in D$ 、 $B \in D$ ，则 **AJB** 为交集型歧义字段。此时，**AJB** 有 AJ/B 、 A/JB 两种切分形式。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 $AB \in D$ 、 $A \in D$ 、 $B \in D$ ，则 **AB** 为组合型歧义字段。此时，**AB** 有 AB 、 A/B 两种切分形式。

- 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

• 3. 混合型歧义

- 同时包含交集型歧义和组合型歧义的歧义字段 人才能
- 这样的/人才/能/经受住考验
- 这样的/人/才/能/经受住考验
- 这样的/人/才能/经受住考验
- 交集型歧义、组合型歧义分布
- 中文文本中交集型切分歧义与组合型切分歧义的 出现比例
约为1:22[1]

[1]刘挺、王开铸，1998，关于歧义字段切分的思考与实验。《中文信息学报》
第2期，63-64页。

切分歧义

- 交集型歧义的链长

- 交集型歧义字段中含有交集字段的个数，称为链长。
- 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

切分歧义

- 真实文本中交集型歧义字段分布[1]。(510万新闻语料)

链长	1	2	3	4	5	6	7	8	总计
词次数	47402	28790	1217	608	29	19	2	1	78248
比例	50.58	47.02	1.56	0.78	0.04	0.02	0.00	0.00	100
字段数	12686	10131	743	324	22	5	2	1	23914
比例	53.05	42.36	3.11	1.35	0.09	0.02	0.01	0.01	100

[1] 中文文本自动分词和标注，刘开瑛著，商务印书馆，2000，66~67

歧义的分类

1. 真歧义

- 歧义字段在不同的语境中确实有多种切分形式
- 地面积
 - 这块/地/面积/还真不小 地面/积/了厚厚的雪
- 和平等
 - 让我们以爱心/和/平等/来对待动物
 - 阿美首脑会议将讨论巴以/和平/等/问题
- 把手
 - 锌合金/把手/的相关求购信息 别/把/手/伸进别人的口袋里

歧义的分类

• 2. 伪歧义

- 歧义字段单独拿出来看有歧义，但在(所有)真实语境中仅有
一种切分形式可接受。
- 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

歧义的分类

根据文献[1]，对于交集型歧义字段，真实文本中伪歧义现象远远多于真歧义现象。

- 伪歧义 94%
- 真歧义 6%
 - ◆ 多种切分形式均匀分布 12%
 - 应用于
 - 将信息技术/应用/于/教学实践
 - 信息技术/应/用于/教学中的哪个方面
 - ◆ 一种切分形式占优 88%
 - 解除了
 - 上级/解除/了/他的职务(大多数)
 - 方程的/解/除了/零以外还有...

[1] 中文文本自动分词和标注，刘开瑛著，商务印书馆，2000，66~67

歧义的发现

- 歧义消解的前提是发现歧义。切分算法应该有能力检测到输入文本中何处出现了歧义切分现象。
- MM和RMM法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)
 - 同时采用MM法和RMM法
 - 如果MM法和RMM法给出同样的结果，则认为没有歧义，若不同，则认为发生了歧义。
 - 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，存在盲点
 - ◆ 最大匹配法不能发现组合型歧义（长词优先）
 - 输入：他从马上下来
MM、RMM：他/从/马上/下来
 - ◆ 在一定条件下（链长为偶数），双向最大匹配法也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

歧义的发现

- 统计数据[1]
 - ◆ 文本中90%左右的句子，MM和RMM结果相同且正确。
 - ◆ 文本中1%左右的句子，MM和RMM结果相同且不正确。
 - ◆ 文本中9%左右的句子，MM和RMM结果不相同（其中一个正确或两者均不正确）（检测到歧义）
- 双向最大匹配法使用较为广泛的原因。

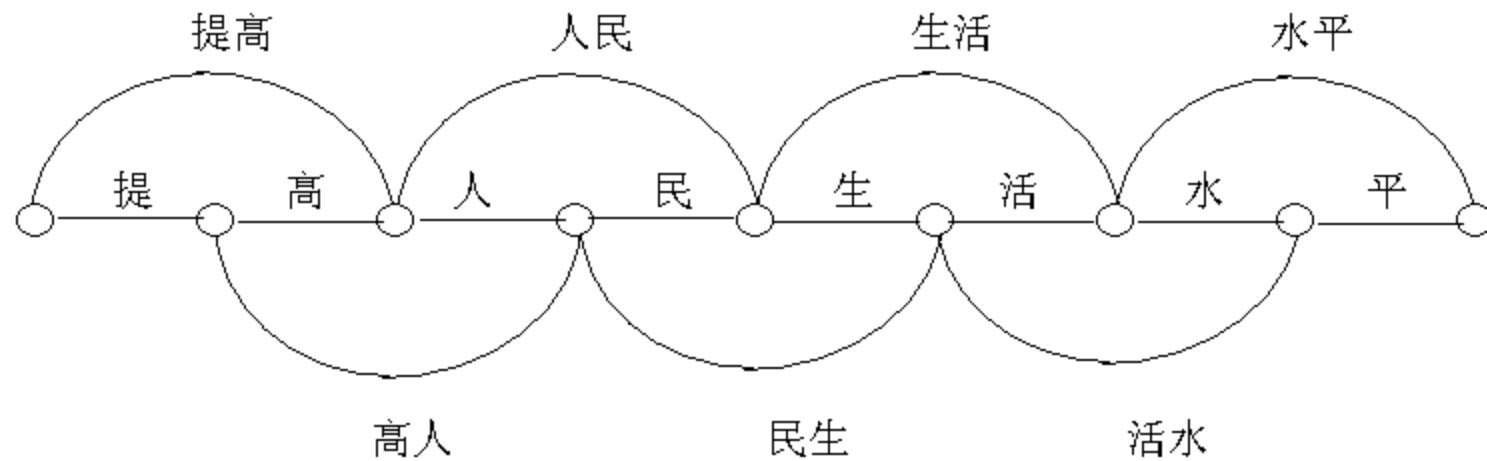
[1] Sun,M.S.and Benjamin K. T. 1995. Ambiguity resolution in Chinese word segmentation. Proceedings of the 10th Asia Conference on Language,Information and Computation, 121 -126.Hong Kong.

歧义的发现

- MM+逆向最小匹配法
- 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

- 基于记忆的歧义消解
 - 伪歧义所占比例非常大
 - 文献[1]从一个1亿字真实汉语语料库中抽取出的前4619个高频交集型歧义切分覆盖了该语料库中全部交集型歧义切分的59.20%，其中4279个属伪歧义，覆盖率高达53.35%。鉴于伪歧义的消解与上下文无关，对伪歧义型高频交集型歧义切分，可以把它们的正确（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。

[1]孙茂松、左正平等，1999，高频最大交集型歧义切分字段在汉语自动分词中的作用。《中文信息学报》第1期，27-34页。

歧义消解

- 基于规则的歧义消解
 - $P[+R+M+Q+A|Z]+$ ”马上” \rightarrow 马 + 上
他从大红 / 马 / 上 / 下来
这件事需要 / 马上 / 办
 - “一起” $+ \sim V \rightarrow$ 一 + 起
 - 我们 / 一起 / 去故宫
 - 一 / 起 / 恶性交通事故

歧义消解

- 基于统计的歧义消解
 - 在词图上寻找统计意义上的最佳路径
- 如何评价最佳路径
- 例如（基于一元模型进行评价）
 - 统计词表中每个词的词频，并将其转换为路径代价
 - ◆ $C = -\log(f/N)$
 - 切分路径的代价为路径上所有词的代价之和
 - 寻求代价最小的路径

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：卡拉OK 波波族 美刀 港刀

未登录词识别

- 未登录词识别困难
 - 未登录词没有明确边界
 - 许多未登录词的构成单元本身都可以独立成词
- 通常，每一类未登录词都要构造专门的识别算法
 - 在序列标注法中，未登录词无需单独处理。
- 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

- 未登录词识别进展
 - 较成熟
 - -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，陈，李，张，刘”等5个大姓覆盖率达32%，姓氏频度表中的前14个高频率的姓氏覆盖率为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%

中文人名识别

- 一个识别模型
 - $r_1: \text{word} \rightarrow \text{name}$
 - $r_2: \text{name} \rightarrow 1\text{-hanzifamily } 2\text{-hanzigiven}$
 - $r_3: \text{name} \rightarrow 1\text{-hanzifamily } 1\text{-hanzigiven}$
 - $r_4: \text{name} \rightarrow 2\text{-hanzifamily } 2\text{-hanzigiven}$
 - $r_5: \text{name} \rightarrow 2\text{-hanzifamily } 1\text{-hanzigiven}$
 - $r_6: 1\text{-hanzifamily} \rightarrow \text{hanzi}_i$
 - $r_7: 2\text{-hanzifamily} \rightarrow \text{hanzi}_i \text{ hanzi}_j$
 - $r_8: 1\text{-hanzigiven} \rightarrow \text{hanzi}_i$
 - $r_9: 2\text{-hanzigiven} \rightarrow \text{hanzi}_i \text{ hanzi}_j$

中文人名识别

- 计算一个可能的人名字串的概率，若其概率大于某个阈值，则判断为人名。

$$\begin{aligned} & P(C_1 C_2 C_3) \\ &= P(r_1) \cdot P(r_2) \cdot P(r_6) \cdot P(r_9) \\ &= P(name) \cdot P(1\text{-hanzifamily } 2\text{-hanzigiven} \mid name) \\ &\quad \cdot P(C_1 \mid 1\text{-hanzifamily}) \cdot P(C_2 C_3 \mid 2\text{-hanzigiven}) \end{aligned}$$

评测

- 国内863、973、中文信息学会
- 国际ACL SIGHAN bakeoff (2003~2007)

Site	word count	R	c_p	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。
- 词就是说话的时候表示思想中一个观念的词。
- 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个人工切分，两两比较

[1] 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）
- 什么是切分单位和应用有关
- 工程观点
- 《资讯处理用中文分词规范》台湾中研院，1995

阅读文献

- [1] 汉语自动分词研究评述
- [3] A Stochastic Finite State Word Segmentation Algorithm for Chinese



词向量概述

How do we represent the meaning of a word?

定义：含义 (meaning)

- 用一个词、词组等表示的概念
- 一个人想用语言、符号等来表达的想法
- 表达在作品、艺术等方面的思想

从语言方式 (linguistic way) 来看含义 (meaning)：语言符号与语言意义 (想法、事情) 的相互对应

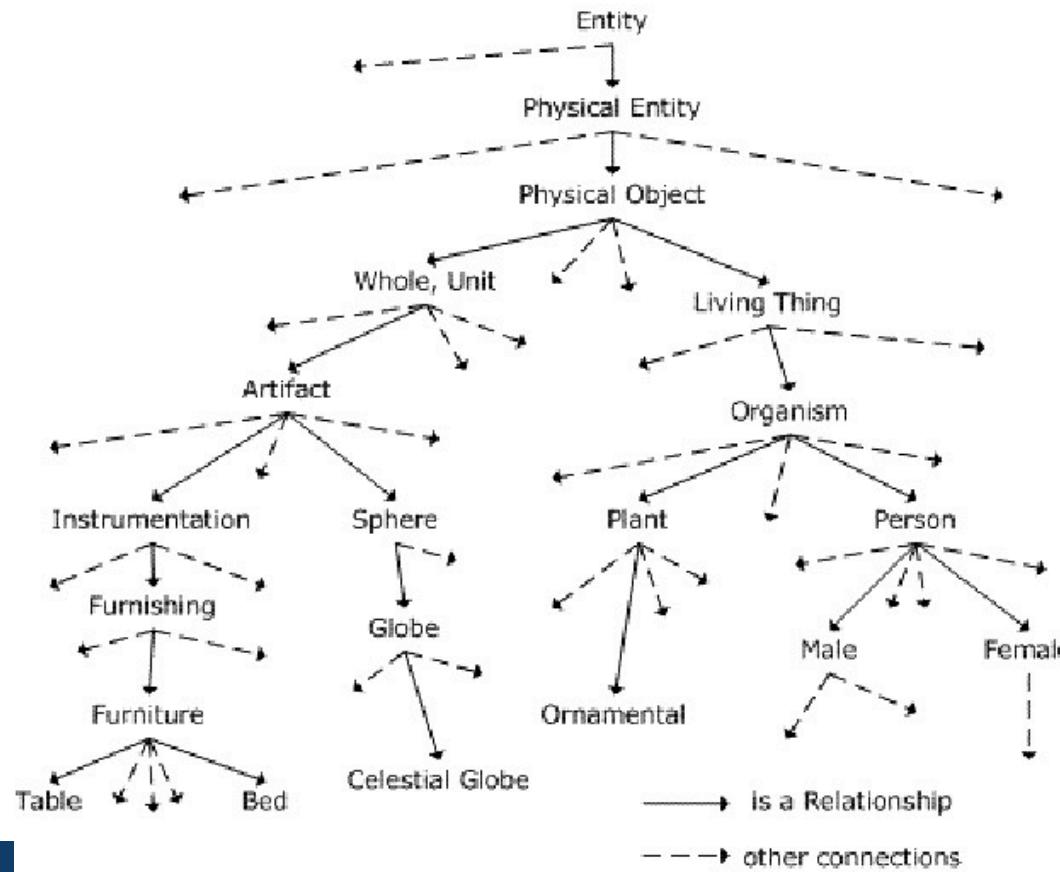
signifier (symbol) \Leftrightarrow signified (idea or thing)

= denotational semantics

tree $\Leftrightarrow \{ \text{🌳}, \text{🌲}, \text{🌴}, \dots \}$

How do we have usable meaning in a computer?

- 要使用计算机处理文本词汇，一种处理方式是**WordNet**: 即构建一个包含同义词(synonym)集和上位词(hypernym)（“is a”关系）的列表的辞典。



WordNet?

e.g., synonym sets containing “good”:

```
from nltk.corpus import wordnet as wn
poses = { 'n':'noun', 'v':'verb', 's':'adj (s)', 'a':'adj', 'r':'adv'}
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
        ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

e.g., hypernyms of “panda”:

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(pandaclosure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

Problems with resources like WordNet

- 忽略了词汇的细微差别
 - 例如“proficient”被列为“good”的同义词。这只在某些上下文中是正确的。
- 缺少单词的新含义
 - 难以持续更新！
 - 例如：wicked、badass、nifty、wizard、genius、ninja、bombast
- 因为是小部分专家构建的，有一定的主观性
- 构建与调整都需要很多的人力成本
- 无法定量计算出单词相似度

Representing words as discrete symbols

- 在传统的自然语言处理中，我们会把词语看作离散的符号：例如 hotel、conference、motel 等。
- 一种文本的离散表示形式是把单词表征为 one-hot 向量的形式
 - One-hot：只有一个1，其余均为0的稀疏向量
- 在 one-hot 向量表示中，向量维度 = 词汇量（如 500,000），以下为一些 one-hot 向量编码过后的单词向量示例：

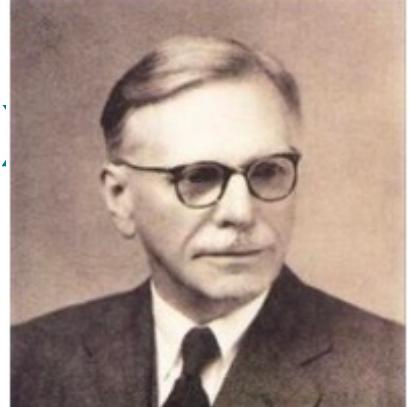
motel = [0 0 0 0 0 0 0 0 0 1 0 0 0 0]

hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0]

Problem with words as discrete symbols

- 例子：用户搜索“Seattle motel”，我们想要找到包含“Seattle hotel”的文档，
- 然而：
 $\text{motel} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]$
 $\text{hotel} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$
- 所有词向量是正交的。对于one-hot向量，没有关于相似性概念。
- 解决思路：
 - ① 使用类似WordNet的工具中的同义词表？
 - 会因不够完整而失败
 - ② 通过大量数据学习词向量本身相似性，获得更精确的稠密词向量编码

Representing words by their context



- **分布式语义：**一个单词的意思是由经常出现在它附近的单词给出的

英国语言学家 J.R. Firth

- “You shall know a word by the company it keeps” (J. R. Firth 1957.11)
- 这是现代统计NLP最成功的理念之一，总体思路有点物以类聚，人以群分的感觉
- 当一个单词w出现在文本中时，它的上下文是出现在其附近的一组单词(在一个固定大小的窗口中)
- 基于海量数据，使用w的许多上下文来构建w的表示
- 如图所示，banking的含义可以根据上下文的内容来表示

...government debt problems turning into banking crises as happened in 2009...
...saying that Europe needs unified banking regulation to replace the hodgepodge...
...India has just given its banking system a shot in the arm...



These context words will represent **banking**

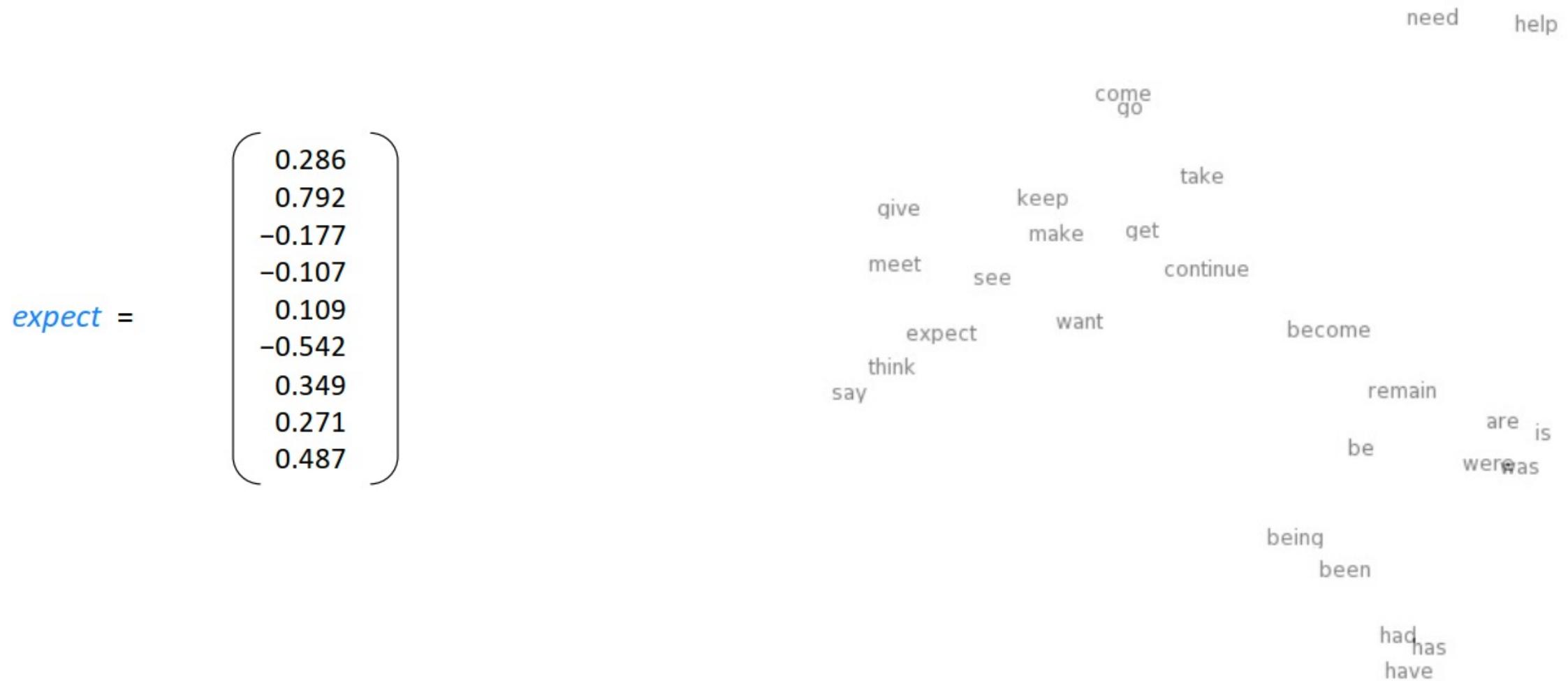
词向量

- 下面我们要介绍词向量的构建方法与思想，我们希望为每个单词构建一个稠密表示的向量，使其与出现在相似上下文中的单词向量相似。

$$\begin{aligned} \text{banking} &= \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix} \\ \text{monetary} &= \begin{pmatrix} 0.413 \\ 0.582 \\ -0.007 \\ 0.247 \\ 0.216 \\ -0.718 \\ 0.147 \\ 0.051 \end{pmatrix} \end{aligned}$$

- 词向量 (word vectors) 有时被称为词嵌入 (word embeddings) 或词表示 (word representations)，是一种分布式表示 (distributed representation)。

Word meaning as a neural word vector – visualization



Word2vec: Overview

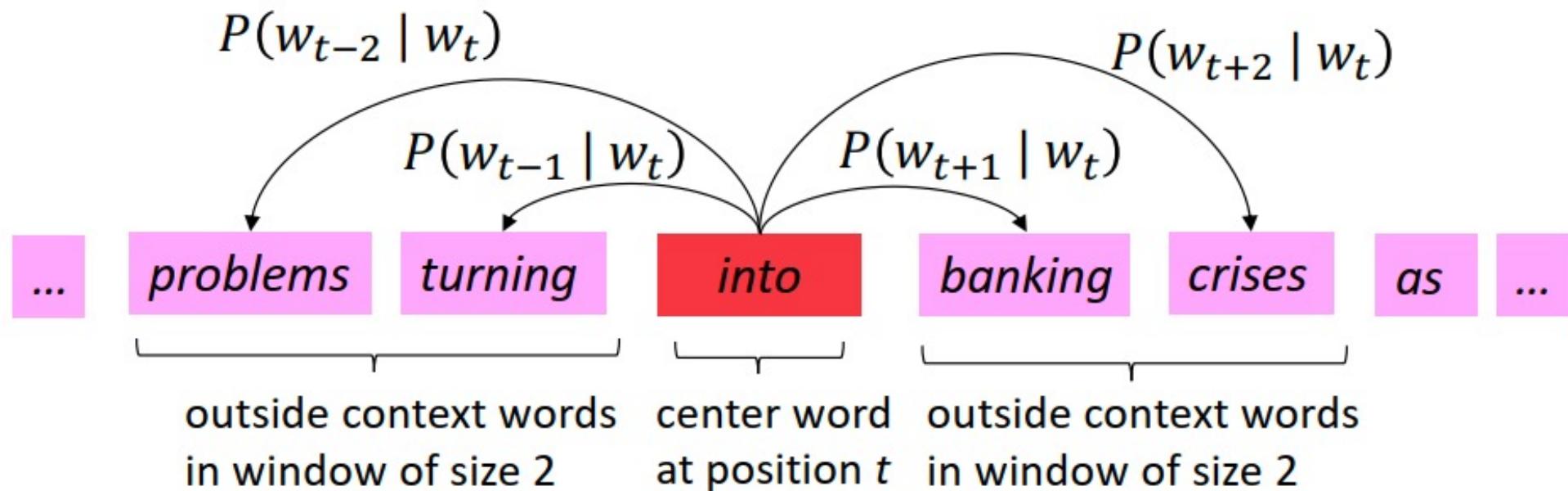
Word2vec (Mikolov et al. 2013)是一个学习词向量表征的框架。

Idea:

- We have a large corpus (“body”) of text: a long list of words
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word c and context (“outside”) words o
- Use the similarity of the word vectors for c and o to calculate the probability of o given c $p(o|c)$ (or vice versa $p(c|o)$)
- Keep adjusting the word vectors to maximize this probability

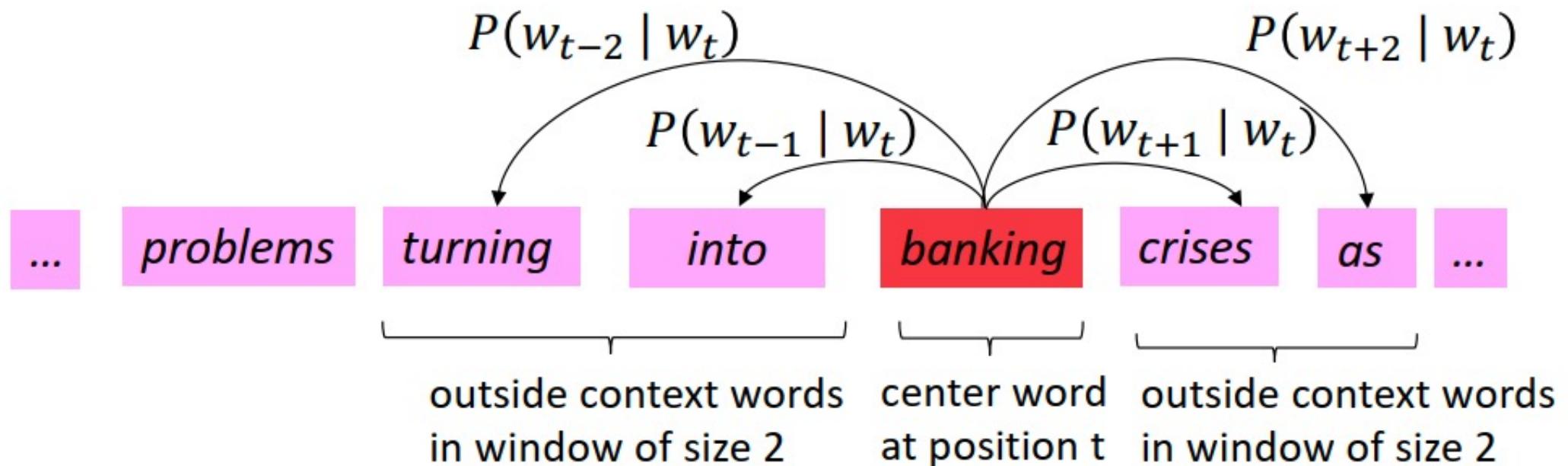
Word2vec: Overview

Example windows and process for computing $P(w_{t+j} | w_t)$



Word2Vec Overview

Example windows and process for computing $P(w_{t+j} | w_t)$



Word2vec: objective function

- 对于每个位置 $t = 1, \dots, T$, 在大小为 m 的固定窗口内预测上下文单词, 给定中心词 w_t , 极大似然函数可以表示为:

$$Likelihood = L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} | w_t; \theta)$$

θ 代表所有要优化的变量

- 对应上述似然函数的目标函数 (objective/cost/loss function) $J(\theta)$ 可以取作 (平均) 负对数似然

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

Minimizing objective function \Leftrightarrow Maximizing predictive accuracy

Word2vec: objective function

- We want to minimize the objective function

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

- Question: How to calculate $P(w_{t+j} | w_t; \theta)$?

- Answer: We will use two vectors per word w:

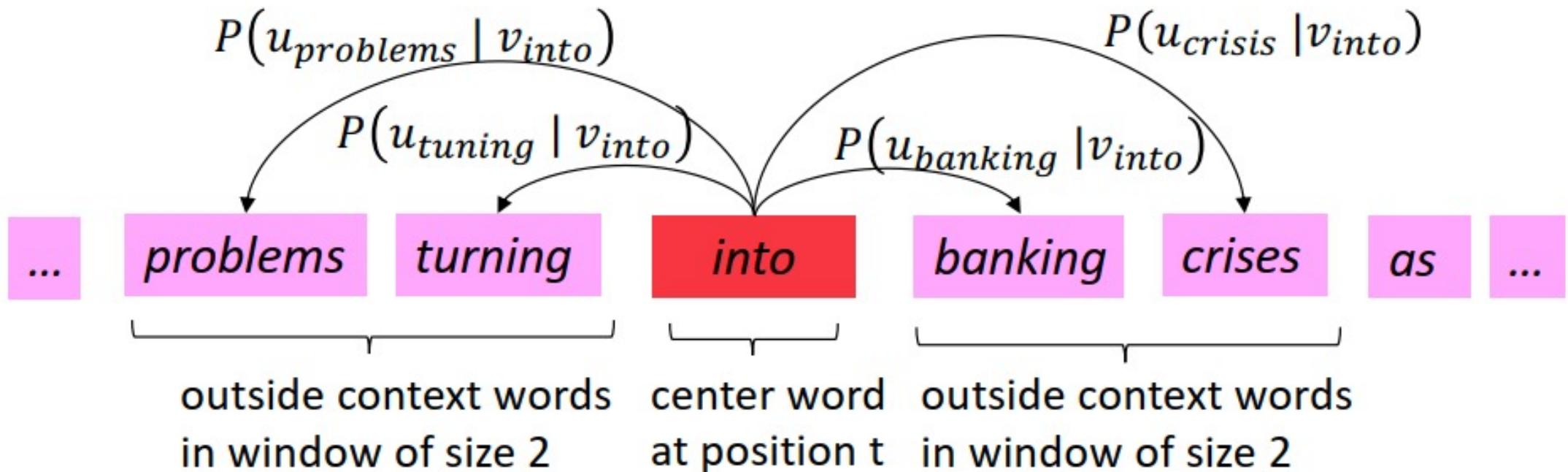
- v_w when w is a center word
- u_w when w is a context word

- Then for a center word c and a context word o:

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Word2Vec with Vectors

- 下图为计算 $P(w_{t+j} | w_t)$ 的示例，这里把 $P(problems | into ; u_{problems}, v_{into}, \theta)$ 简写为 $P(u_{problems} | v_{into})$ ，例子中的上下文窗口大小2，即“左右2个单词+一个中心词”。



Word2vec: prediction function

② Exponentiation makes anything positive

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

① Dot product compares similarity of o and c .
 $u^T v = u \cdot v = \sum_{i=1}^n u_i v_i$
Larger dot product = larger probability

③ Normalize over entire vocabulary
to give probability distribution

- Softmax 函数举例: $\mathbb{R}^n \rightarrow (0,1)^n$

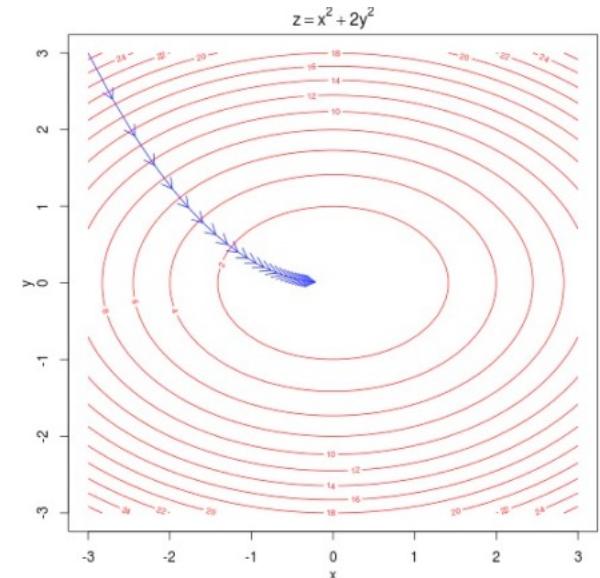
$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} = p_i$$

- Softmax函数将任意实数向量 x_i ($i=1, \dots, n$) 映射为一个概率分布 p_i ($i=1, \dots, n$)
 - Max: 最大元素拥有最大的概率值
 - Soft: 小的元素依然拥有概率值, 不会是0
 - 在深度学习中普遍应用

To train the model: Optimize value of parameters to minimize loss

- 模型训练过程中，我们需要逐步调整参数来最小化loss
- θ 代表所有模型的参数
- 此处，我们每个词向量长度为d，一共V个词，那么我们有→
- 每个词有两个向量
- 整个优化过程顺着梯度的方向下降，直到找到最低点
- 所有向量的梯度都要计算！

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$



具体的loss function求梯度的流程

$$\min J(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w'_{t+j} | w_t)$$

↑ ↑
文本长度 窗口长度

负对数似然

此处：

$$p(o|c) = \frac{\exp(u_o^\top v_c)}{\sum_{w=1}^V \exp(u_w^\top v_c)}$$

接下来，看看梯度怎么求。

具体的loss function求梯度的流程

此处是对向量求偏导

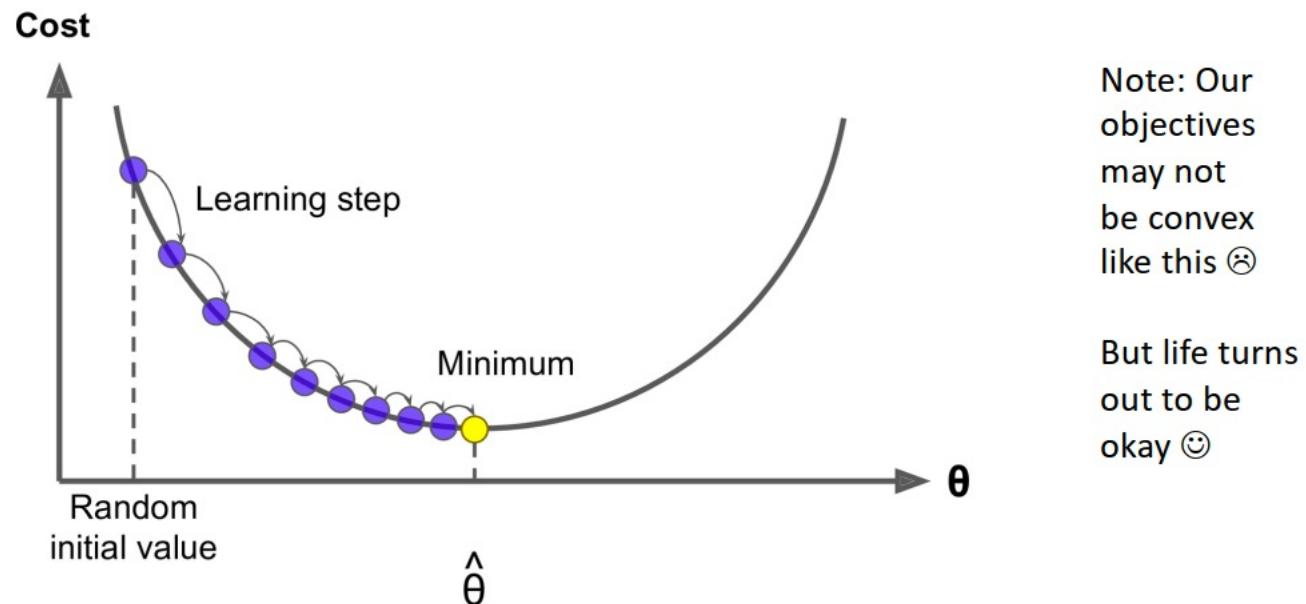
$$\begin{aligned}\frac{\partial}{\partial v_c} \log P(o|c) &= \frac{\partial}{\partial v_c} \log \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)} \\ &= \frac{\partial}{\partial v_c} \left(\log \exp(u_o^T v_c) - \log \sum_{w \in V} \exp(u_w^T v_c) \right) \\ &= \frac{\partial}{\partial v_c} \left(u_o^T v_c - \log \sum_{w \in V} \exp(u_w^T v_c) \right) \\ &= u_o - \frac{\sum_{w \in V} \exp(u_w^T v_c) u_w}{\sum_{w \in V} \exp(u_w^T v_c)} \\ &= u_o - \sum_{w \in V} \frac{\exp(u_w^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)} u_w \\ \text{Observed - expected} &= u_o - \boxed{\sum_{w \in V} P(w|c) u_w}\end{aligned}$$

• 这一段是对中心向量 v_c 的梯度求法
• 输出向量 u_o 的求法类似

把所有上下文向量用其概率加权求和

Optimization: Gradient Descent

- 我们现在需要最小化
- 利用梯度下降法
- 基本思路：对于当前的参数 θ ，计算 $J(\theta)$ 的梯度 $\nabla J(\theta)$ ，然后在负梯度的方向不断地小步迭代



Gradient Descent

- 更新方程

$$\theta^{new} = \theta^{old} - \alpha \nabla_{\theta} J(\theta)$$

α = step size or learning rate

- Python代码

```
while True:  
    theta_grad = evaluate_gradient(J,corpus,theta)  
    theta = theta - alpha * theta_grad
```

Stochastic Gradient Descent

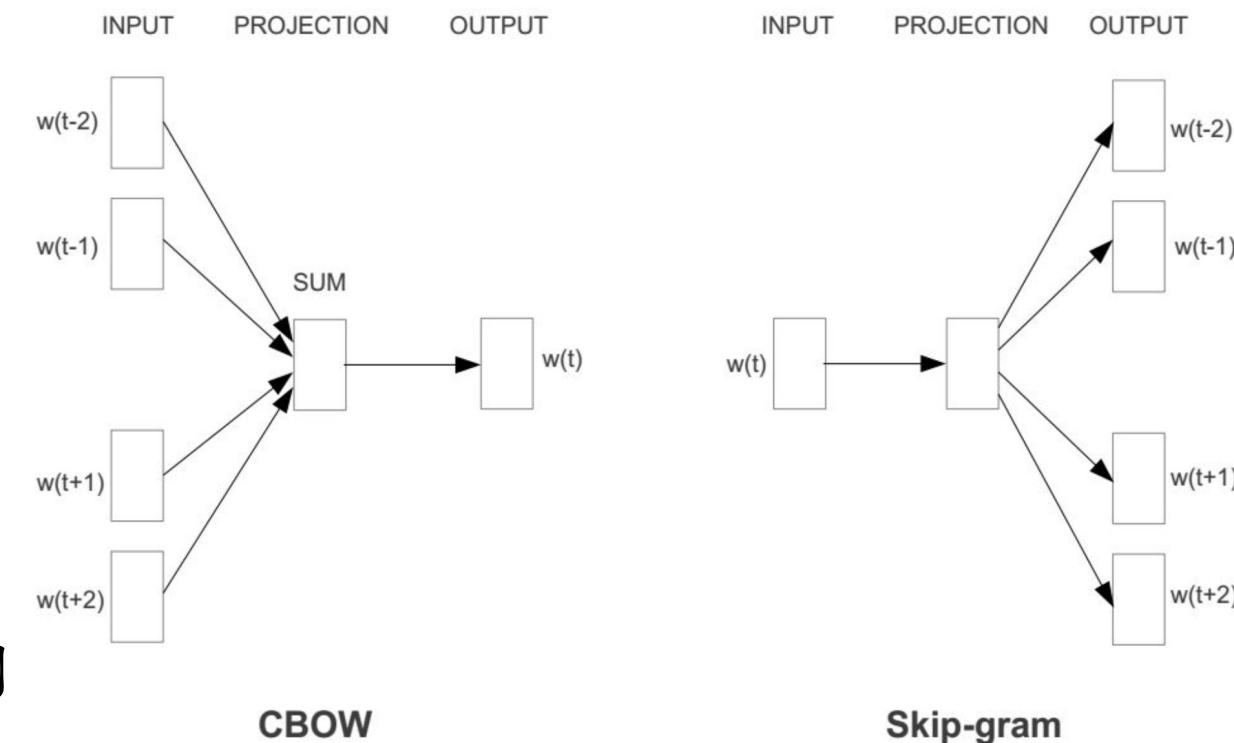
- 不足之处： $J(\theta)$ 是语料库中所有窗口的函数（上亿）
 - $\nabla J(\theta)$ 的计算需要花费大量资源
- 对于任意神经网络都不是好主意
- 解决方案：Stochastic gradient descent (SGD) 随机梯度下降
 - 不断的采样窗口，每采样一次做一次更新
- 代码示例

```
while True:  
    window = sample_window(corpus)  
    theta_grad = evaluate_gradient(J,window,theta)  
    theta = theta - alpha * theta_grad
```

Word2vec algorithm family: More details

- 为什么每个单词一定要对应两个向量?
 - 最终两个向量的平均值代表词向量
 - 实际操作中, 每个词只对应一个向量。。。而且效果更好一些。
- 其他可能的模型结构
 - Skip-gram: 利用中心词预测上下文词 $p(o|c)$
 - Continuous Bag of Words(CBOW): 利用上下文词预测中心词 $p(c|o)$

我们刚刚讲过的是skip-gram model



Training efficiency

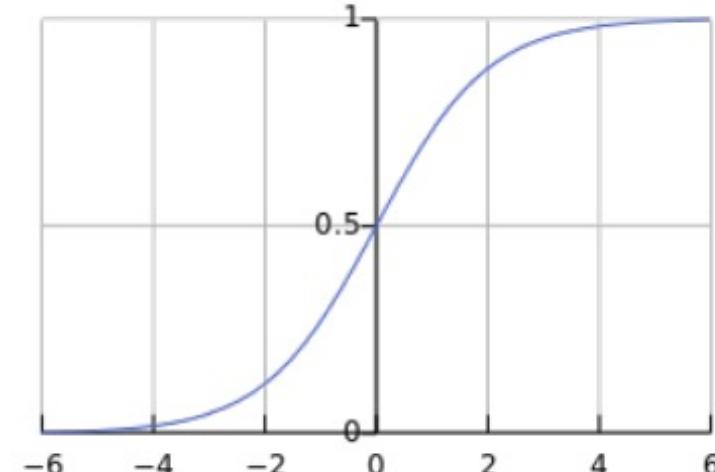
- 为了提高训练效率
 - Hierarchical Softmax (在算力足够的条件下不实用, 略)
 - Negative sampling
- Why? 归一化项的计算太费资源
$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$
- 所以在Word2vec的标准实现中用的是负采样方法
- 主要思想: 使用一个 true pair (中心词及其上下文窗口中的词)与几个 noise pair (中心词与随机词搭配)形成的样本, 训练二元逻辑回归。

The skip-gram model with negative sampling

- From paper: “Distributed Representations of Words and Phrases and their Compositionality” (Mikolov et al. 2013)
- 目标函数（最大化） $J(\theta) = \frac{1}{T} \sum_{t=1}^T J_t(\theta)$

$$J_t(\theta) = \log \sigma(u_o^T v_c) + \sum_{i=1}^k \mathbb{E}_{j \sim P(w)} [\log \sigma(-u_j^T v_c)]$$

- Sigmoid函数 $\sigma(x) = \frac{1}{1+e^{-x}}$
- 我们要最大化2个词共现的概率，
最小化与噪音词的共现概率



The skip-gram model with negative sampling

- 实际操作中：

$$J_{\text{neg-sample}}(\mathbf{u}_o, \mathbf{v}_c, U) = -\log \sigma(\mathbf{u}_o^T \mathbf{v}_c) - \sum_{k \in \{K \text{ sampled indices}\}} \log \sigma(-\mathbf{u}_k^T \mathbf{v}_c)$$

- 我们取 k 个负例采样
- 最大化出现在窗口中的“中心词”之外的词语出现的概率，而最小化其他没有出现在窗口中的随机词的概率
- 对词进行采样的时候使用概率分布： $P(w) = U(w)^{3/4} / Z$ ，其中 $U()$ 代表一元的分布 (w 出现的概率)
- $3/4$ 次幂减少了单词出现频率之间的差异，从而提高了低频词被抽中的概率

Why not capture co-occurrence counts directly?

- 与其一遍一遍的遍历语料库优化模型，为什么不直接统计一下单词与附近单词的共现情况呢？
- 建立共现矩阵 X ：两种形式：window和全文档
 - Window : 与word2vec类似，在每个单词周围都使用Window，捕捉一些语法和语义信息
 - Word-document : 共现矩阵的基本假设是在同一篇文章中出现的单词更有可能相互关联。假设单词 i 出现在文章 j 中，则矩阵元素 X_{ij} 加一，当我们处理完数据库中的所有文章后，就得到了矩阵 X ，其大小为 $|V| \times M$ ，其中 $|V|$ 为词汇量，而 M 为文章数。这一构建单词文章co-occurrence matrix的方法也是经典的Latent Semantic Analysis (LSA) 所采用的

Example: Window based co-occurrence matrix

- Window length 1 (more common: 5–10)
- Symmetric (irrelevant whether left or right context)
- Example corpus:
 - I like deep learning
 - I like NLP
 - I enjoy flying

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Co-occurrence vectors

- 共现向量的问题
 - 用共现次数衡量单词的相似性，但是会随着词汇量的增加而增大矩阵的大小。
 - 需要很多空间来存储这一高维矩阵。
 - 后续的分类模型也会由于矩阵的稀疏性而存在稀疏性问题，使得效果不佳。
- 低维向量
 - 如何降维，获得低维稠密向量？
 - 25-1000维，类似Word2vec？

Classic Method: Dimensionality Reduction on X

- SVD分解 (Singular Value Decomposition)

$$X^k = U \Sigma V^T$$

Retain only k singular values, in order to generalize.

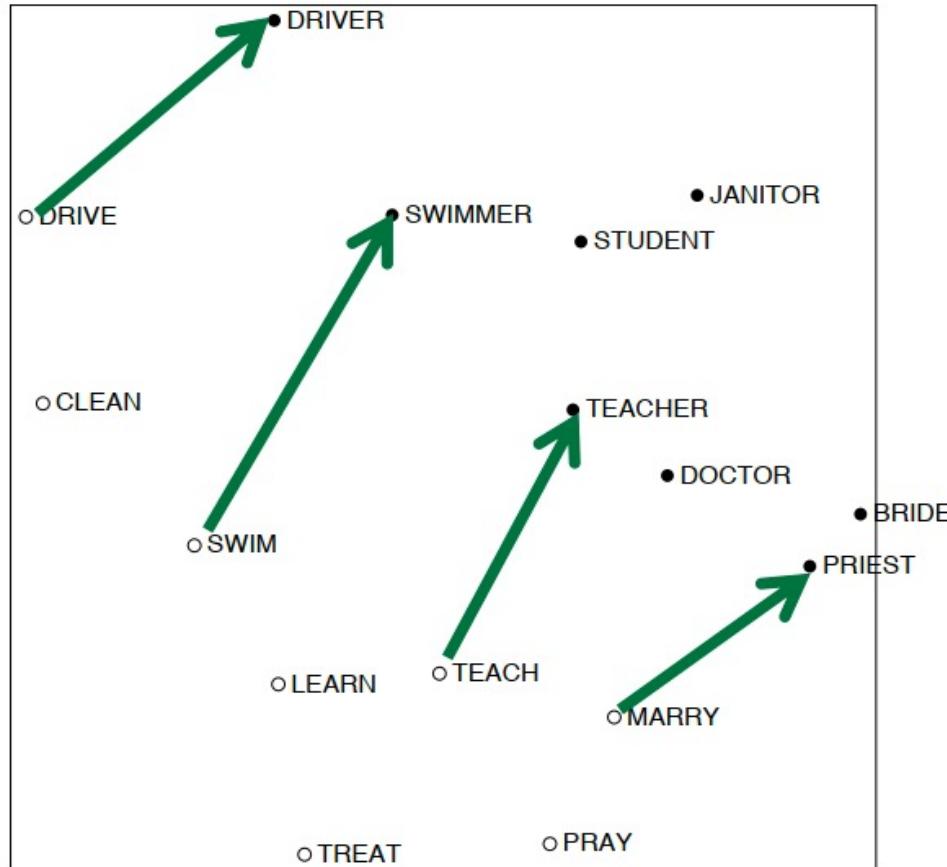
\hat{X} is the best rank k approximation to X , in terms of least squares.

Classic linear algebra result. Expensive to compute for large matrices.

Hacks to X (several used in Rohde et al. 2005 in COALS)

- 在原始计数的矩阵上运行SVD往往不会得到很好的结果！！
- 问题：一些功能性词汇 (the, he, has) 出现的过多，语法有太多的影响
 - 使用log进行缩放
 - 超过某个值（比如100），一律截断
 - 直接全部忽视功能性词汇
- 在基于window的计数中，提高距离近的单词的计数
- 使用Pearson相关系数代替直接计数，负值直接设为0

Interesting semantic patterns emerge in the scaled vectors



COALS model from

Rohde et al. ms., 2005. An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence

Towards GloVe: Count based vs. direct prediction

基于计数：使用整个矩阵的全局统计数据来直接估计

- LSA, HAL (Lund & Burgess),
- COALS, Hellinger-PCA (Rohde et al, Lebret & Collobert)

优点：训练快速；统计学信息高效利用

缺点：主要用于捕捉单词相似性；对大量数据给予比例失调的重视

基于预估模型：定义概率分布并试图预测单词

- Skip-gram/CBOW (Mikolov et al)
- NNLM, HLBL, RNN (Bengio et al; Collobert & Weston; Huang et al; Mnih & Hinton)

优点：提高其他任务的性能；能捕获除了单词相似性以外的复杂的模式

缺点：随语料库增大会增大规模；统计数据的低效使用（采样是对统计数据的低效使用）

Encoding meaning components in vector differences

[Pennington, Socher, and Manning, EMNLP 2014]

Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	$x = \text{solid}$	$x = \text{gas}$	$x = \text{water}$	$x = \text{random}$
$P(x \text{ice})$	large	small	large	small
$P(x \text{steam})$	small	large	large	small
$\frac{P(x \text{ice})}{P(x \text{steam})}$	large	small	~1	~1

Encoding meaning components in vector differences

[Pennington, Socher, and Manning, EMNLP 2014]

Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	$x = \text{solid}$	$x = \text{gas}$	$x = \text{water}$	$x = \text{fashion}$
$P(x \text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(x \text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(x \text{ice})}{P(x \text{steam})}$	8.9	8.5×10^{-2}	1.36	0.96

Encoding meaning components in vector differences

- 问题：
 - 我们如何在词向量空间中以线性含义成分的形式捕获共现概率的比值？
- 解决方案：

- log-bilinear 模型：

$$w_i \cdot w_j = \log P(i|j)$$

- 向量之间作差的话：

$$w_x \cdot (w_a - w_b) = \log \frac{P(x|a)}{P(x|b)}$$

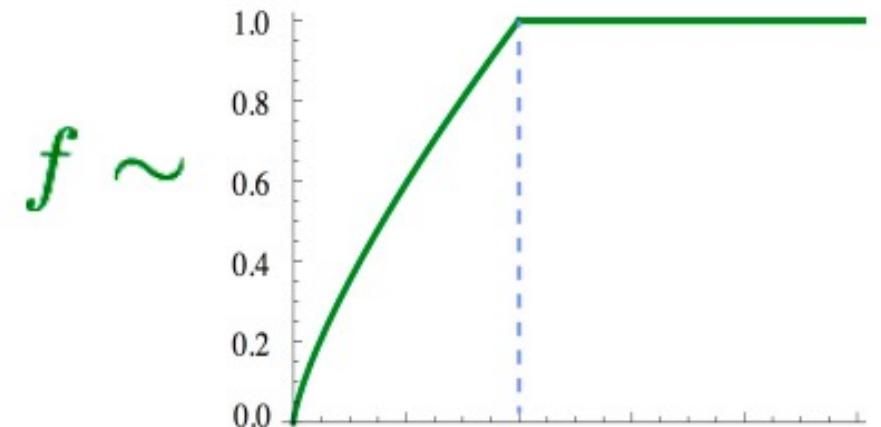
Combining the best of both worlds

GloVe [Pennington, Socher, and Manning, EMNLP 2014]

$$w_i \cdot w_j = \log P(i|j)$$

$$J = \sum_{i,j=1}^V f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

- 训练快速
- 可以扩展到大型语料库
- 即使是小语料库和小向量，性能也很好



GloVe results

Nearest words to
[frog](#):

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus



litoria



rana



leptodactylidae



eleutherodactylus

How to evaluate word vectors?

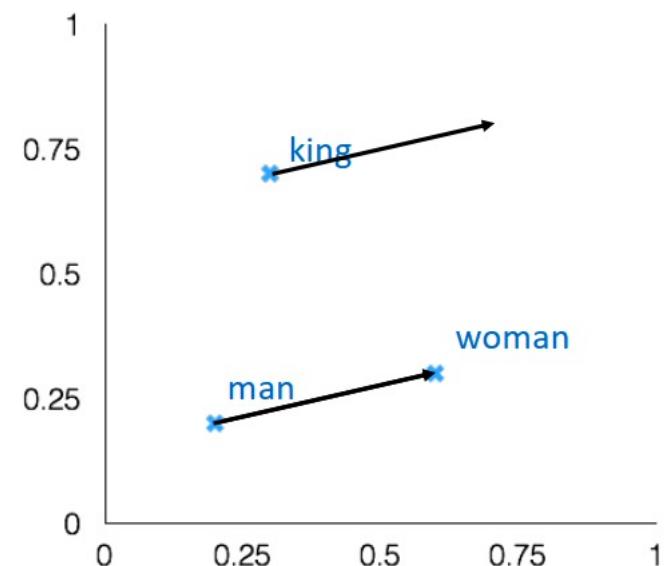
- 我们如何评估词向量呢，有内在和外在两种方式：
- 内在评估方式
 - 对特定/中间子任务进行评估
 - 计算速度快
 - 有助于理解这个系统
 - 不清楚是否真的有用，除非与实际任务建立了相关性
- 外部任务方式
 - 对真实任务（如下游NLP任务）的评估
 - 计算精确度可能需要很长时间
 - 不清楚子系统问题所在，是交互还是其他子系统问题
 - 如果用一个子系统替换另一个子系统可以提高精确度 → winning

Intrinsic word vector evaluation

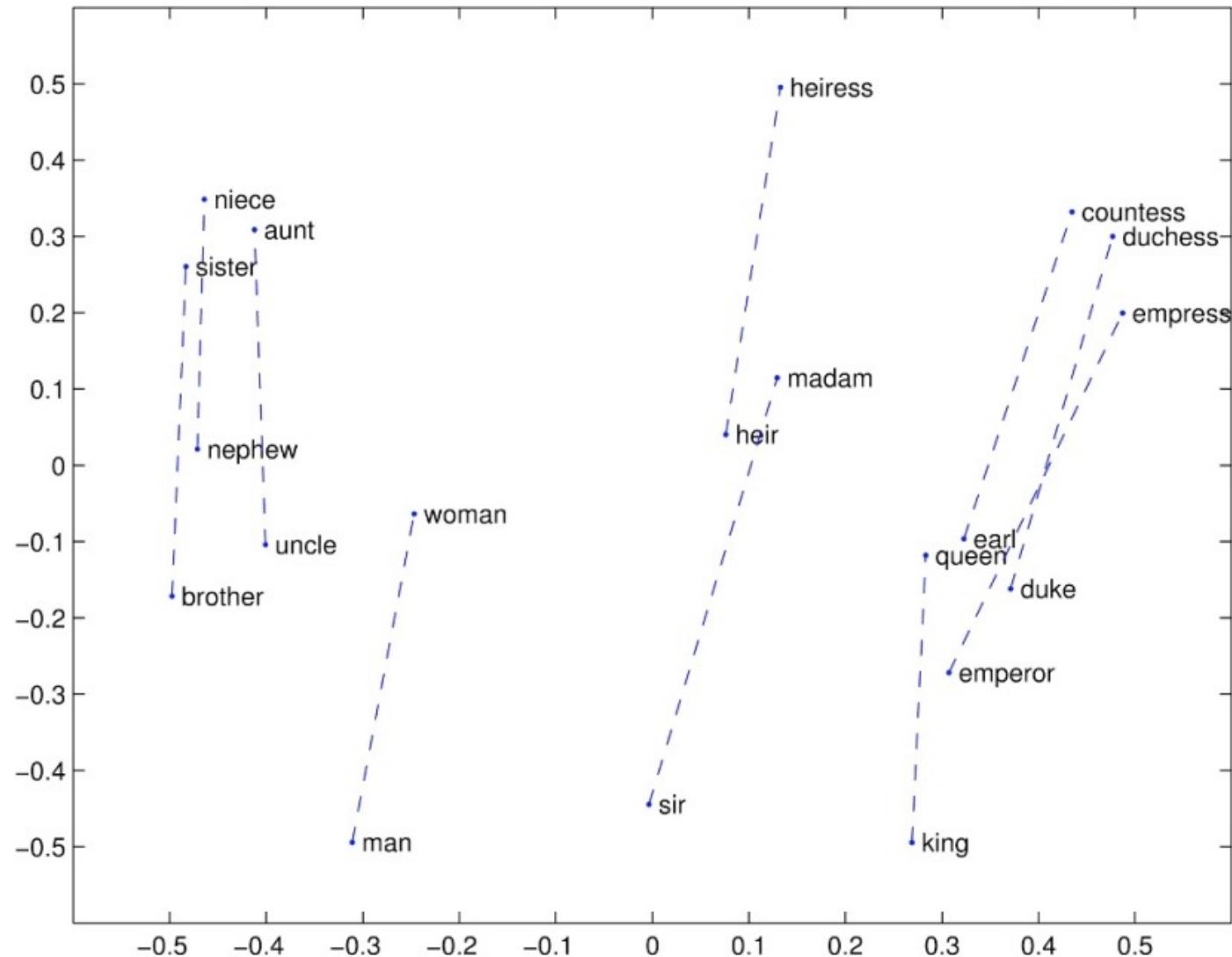
- 一种内在词向量评估方式是“词向量类比”：对于具备某种关系的词对a,b，在给定词c的情况下，找到具备类似关系的词d

$$\begin{array}{c} \boxed{a:b :: c:?} \\ \text{man:woman :: king:?:} \end{array} \longrightarrow \boxed{d = \arg \max_i \frac{(x_b - x_a + x_c)^T x_i}{\|x_b - x_a + x_c\|}}$$

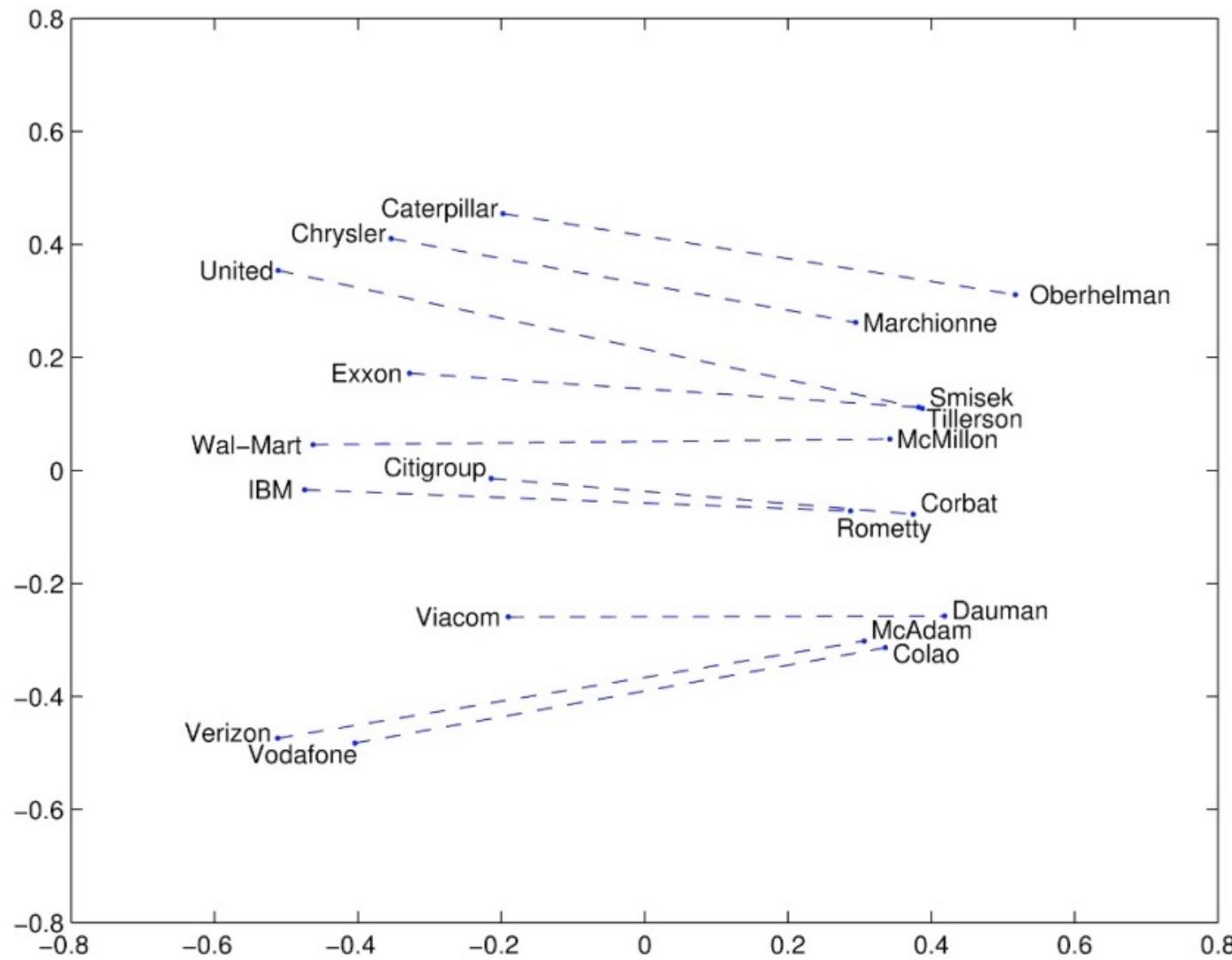
- 通过加法后的余弦距离是否能很好地捕捉到直观的语义和句法类比问题来评估单词向量
- 从搜索中丢弃输入的单词
- 问题：如果有信息但不是线性的怎么办？



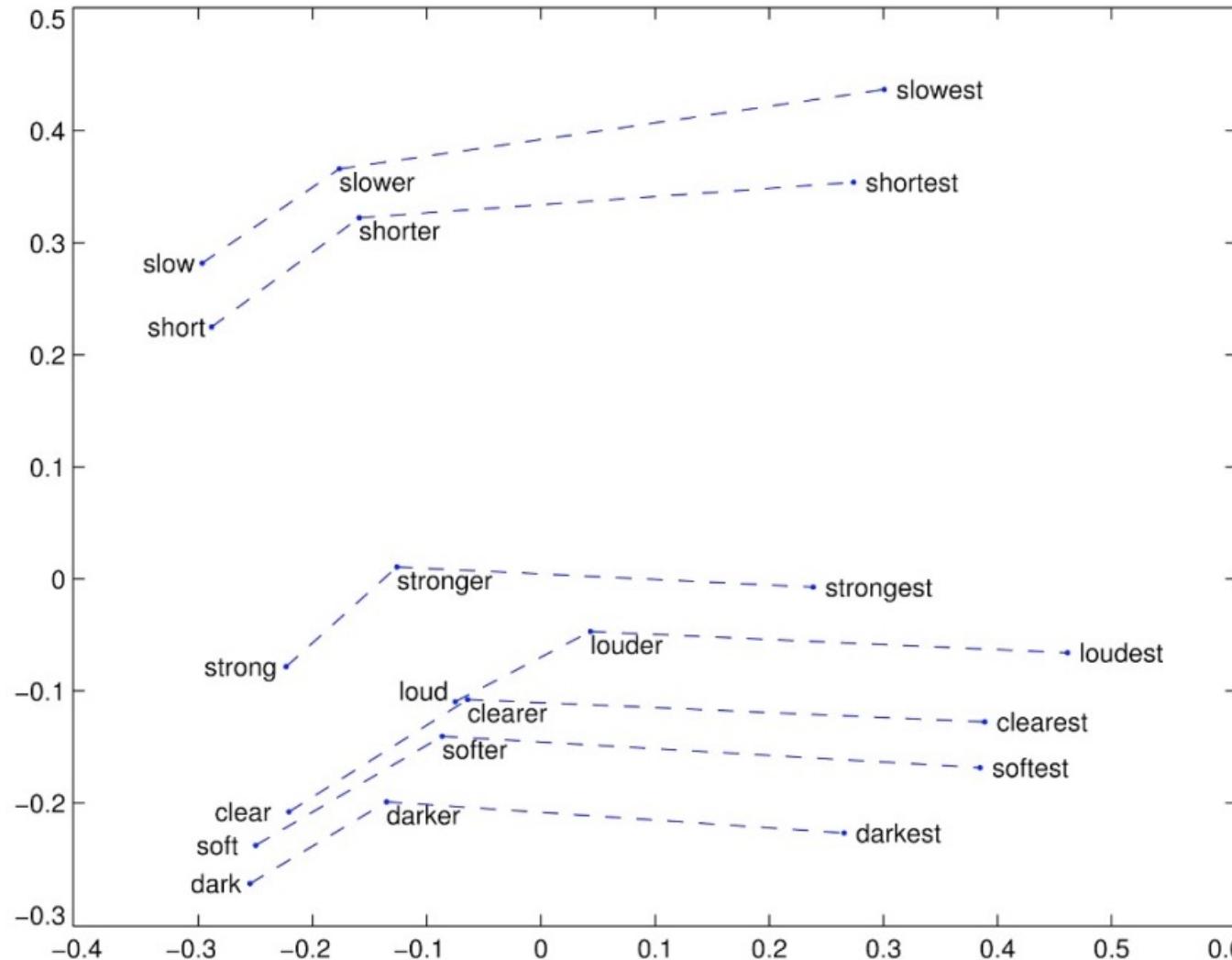
Glove Visualizations



Glove Visualizations: Company - CEO



Glove Visualizations: Comparatives and Superlatives



Analogy evaluation and hyperparameters

- 在一个包含各种关系种类的测试集上测试（包括语法和语义关系）

Glove word vectors evaluation

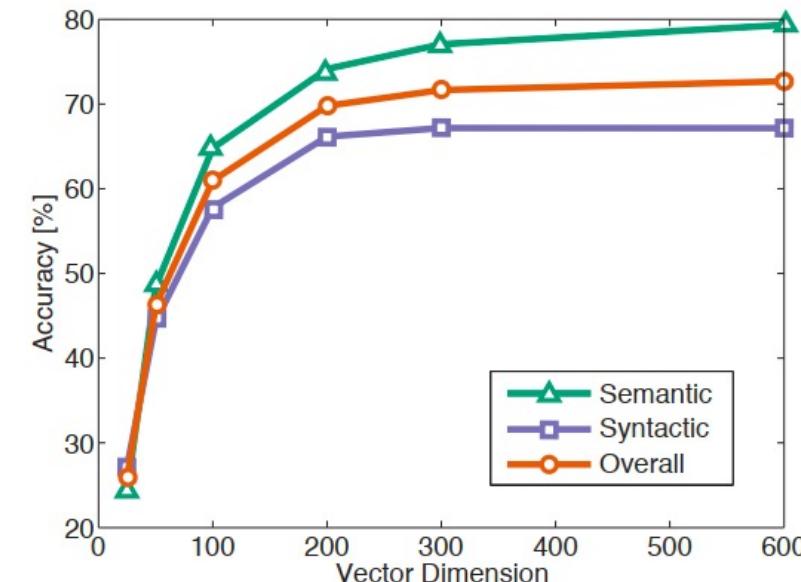
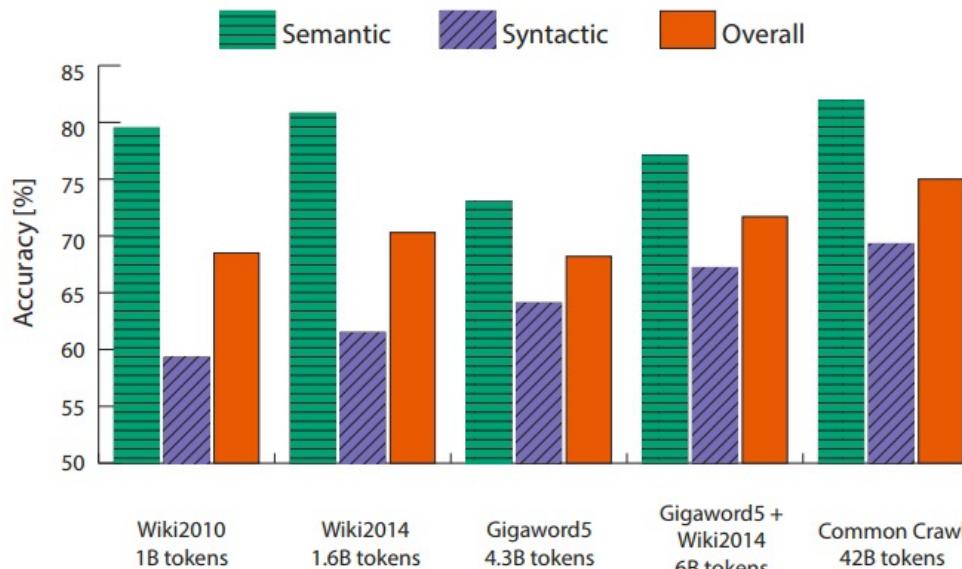
Model	Dim.	Size	Sem.	Syn.	Tot.
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW [†]	300	6B	63.6	<u>67.4</u>	65.7
SG [†]	300	6B	73.0	66.0	69.1
GloVe	300	6B	<u>77.4</u>	67.0	<u>71.7</u>

Analogy evaluation and hyperparameters

- 其他的一些实验发现

- More data helps
- Wikipedia is better than news text!

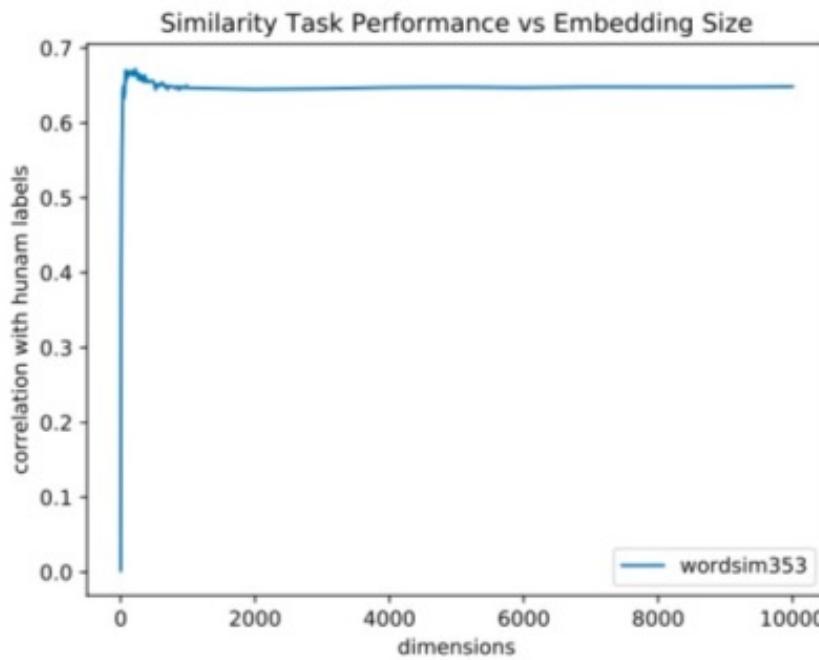
- Dimensionality
- Good dimension is ~ 300



On the Dimensionality of Word Embedding

[Zi Yin and Yuanyuan Shen, NeurIPS 2018]

- <https://papers.nips.cc/paper/7368-on-the-dimensionality-of-word-embedding.pdf>



(b) WordSim353 Test

- 利用矩阵摄动理论，揭示了词嵌入维数选择的基本的偏差与方法的权衡
- 补充说明：当持续增大词向量维度的时候，词向量的效果不会一直变差并且会保持平稳

Another intrinsic word vector evaluation

- 使用 cosine similarity 衡量词向量之间的相似程度并与人类评估比照
- Example dataset: WordSim353
<http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/>

Word 1	Word 2	Human (mean)
tiger	cat	7.35
tiger	tiger	10
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

Extrinsic word vector evaluation

- NER (Named Entity Recognition) : 识别实体中的人名，地名，机构名：Apple locates in Cupertino

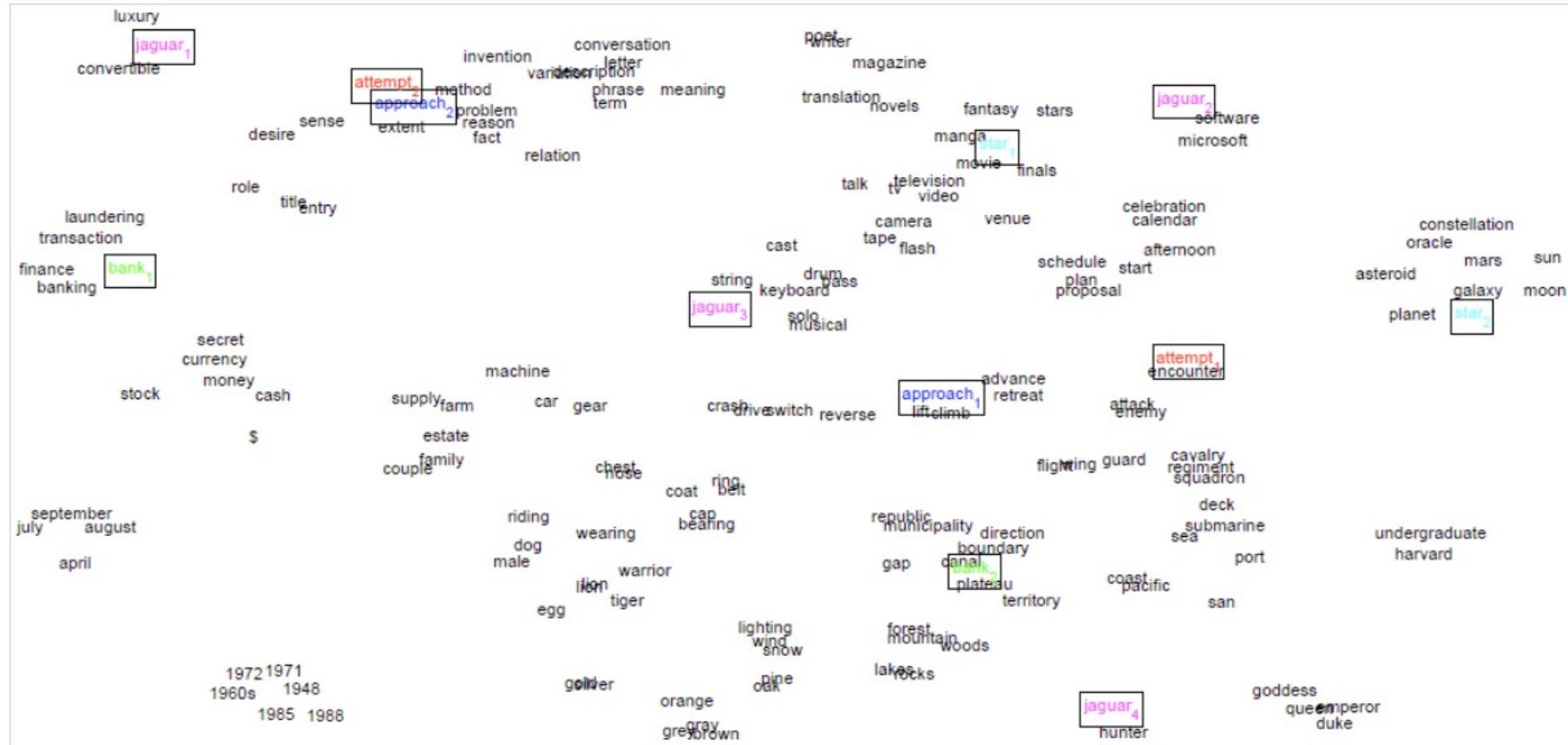
Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	88.7	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	93.2	88.3	82.9	82.2

Word senses and word sense ambiguity

- 大多数单词都是多义的
 - 特别是常见单词
 - 特别是存在已久的单词
- 例如： pike
 - A sharp point or staff 矛
 - A type of elongated fish 梭子鱼
 - A railroad line or system
 - A type of road
 - The future (coming down the pike)
 - A type of body position (as in diving)
 - To kill or pierce with a pike
 - To make one's way (pike along)
 - In Australian English, pike means to pull out from doing something: I reckon he could have climbed that cliff, but he piked!
- 那么，词向量是总体捕捉了所有这些信息，还是杂乱在一起了呢？

Improving Word Representations Via Global Context And Multiple Word Prototypes (Huang et al. 2012)

- 将常用词的所有上下文进行聚类，通过该词得到一些清晰的簇，从而将这个常用词分解为多个单词，例如bank-1, bank-2等



Linear Algebraic Structure of Word Senses, with Applications to Polysemy (Arora, ..., Ma, ..., TACL 2018)

- 此方法中，单词在标准单词嵌入(如word2vec)中的不同含义以线性叠加(加权和)的形式存在

$$v_{\text{pike}} = \alpha_1 v_{\text{pike}_1} + \alpha_2 v_{\text{pike}_2} + \alpha_3 v_{\text{pike}_3} \quad \alpha_1 = \frac{f_1}{f_1 + f_2 + f_3}$$

- f是出现频率
- 结果
 - 只是加权平均值就已经可以获得很好的效果
 - 由于里面用到了稀疏编码的方法，实际上可以将不同词义分离出来
(前提是它们相对比较常见)

tie				
trousers	season	scoreline	wires	operatic
blouse	teams	goalless	cables	soprano
waistcoat	winning	equaliser	wiring	mezzo
skirt	league	clinching	electrical	contralto
sleeved	finished	scoreless	wire	baritone
pants	championship	replay	cable	coloratura

Thank you