Word Embedding
词嵌入简介

数据科学沙龙
李韶华

提纲

- NLP的基本概念
- 词嵌入的直观认识
- 词嵌入的基本原理
- Word2vec的优化
- 一些实验

NLP常见任务

- 自动摘要
- 指代消解　　小明放学了，妈妈去接他
- 机器翻译　　小心地滑 → Slide carefully
- 词性标注　　heat (v.) water (n.) in (p.) a (det.) pot (n.)
- 分词 (中文,日文等)　　大水沟/很/难/过
- 主题识别
- 文本分类
- ……

NLP基本方法

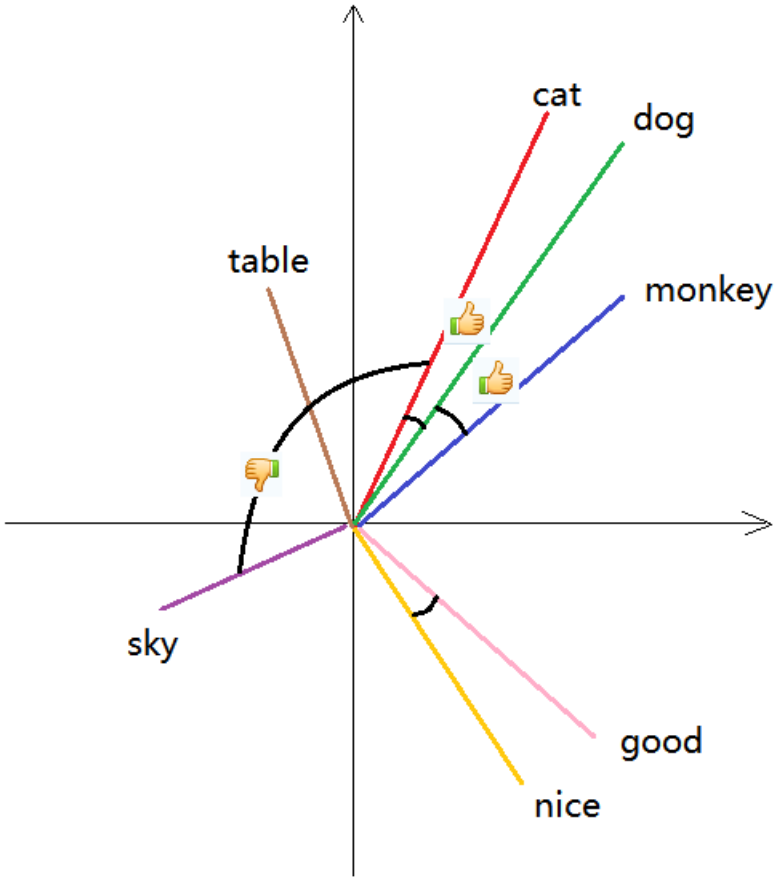
- 传统: 基于规则
- 现代: 基于统计机器学习
  - HMM, CRF, SVM, LDA, CNN…
  - 规则隐含在模型参数里
  - 越来越像机器学习在文本处理方面的应用

One-hot encoding

- 怎么把文本表示成适合机器处理的形式
  - 词是基本单元
  - 之前: 词用one-hot encoding表示
    - “天气”: (1,0,0…,0)，“气候”: (0,1,0,…0)
    - 权力/的/游戏: (1,0,0,1,1,0,0, …)
      冰/与/火/之/歌: (0,1,1,0,0,1,1,…)
- one-hot encoding的问题
  - 高维(几万-几十万)，稀疏
  - 没有编码不同词之间的语义相似性
  - 难以做模糊匹配

词嵌入的直观印象

- 词映射到低维连续向量
  - Cat: (-0.065, -0.035, 0.019, -0.026, 0.085,…)
  - Dog: (-0.019, -0.076, 0.044, 0.021,0.095,…)
  - Table: (0.027, 0.013, 0.006, -0.023, 0.014, …)
- 相似词映射到相似方向
  - 语义相似性被编码了
  - Cosine相似度衡量方向



Projection of the embedding vectors to 2-D

词嵌入可以做类比题

- v(“国王”) – v(“王后”) ≈ v(“男”) – v(“女”)
- v(“英国”) + v(“首都”) ≈ v(“伦敦”)
- 词嵌入编码了语义空间中的线性关系
  - 向量的不同部分对应不同的语义
- 应用
  - 两个句子: A含“英国”,“首都”，不含“伦敦”；B含“伦敦”
  - 所有词的词向量的和表示句子 (“fastText”)
  - 两个句子仍会比较相似

相似词如何映射到相似方向

- 基本假设：“相似”词的邻居词分布类似
- 倒推: 两个词邻居词分布类似 → 两个词语义相近
- 猫 宠物 主人 喂食 蹭 喵
- 狗 宠物 主人 喂食 咬 汪
- ⇨ **v**(“猫”)≈v(“狗”)
- Apple: tree red growth design music company engineering executive
- **v**(“apple”)≈**v**(“orange”), **v**(“apple”)≈**v**(“microsoft”)

词嵌入的优点

- 维度低(100 – 500维), 连续向量, 机器学习算法好处理
- 无监督学习, 容易获得大语料
- 天然有聚类后的效果
- 罕见词也可以学到不错的表示
  - “风姿绰约” ≈ “漂亮”

Word2vec 模型

- 回归连结函数: $$P(w_k|w_i) = \frac{\exp(\tilde{\boldsymbol{v}}_{w_k}^{\top} \boldsymbol{v}_{w_i})}{\sum_{w_j \in V} \exp(\tilde{\boldsymbol{v}}_{w_j}^{\top} \boldsymbol{v}_{w_i})}$$
- 分母: 归一化项，可暂时忽略
- $p(w_k|w_i) \propto \exp(\tilde{\boldsymbol{v}}_{w_k}^{T} \boldsymbol{v}_{w_i})$
- $\tilde{v}_{w_{k,n}}$和$v_{w_{i,n}}$方向相似: 预测的$p(w_k|w_i)$大
  方向不同: 预测的概率小
- w_k经常出现在w_i周围, $p(w_k|w_i)$大, 驱使$\tilde{v}_{w_{k,n}}$和$v_{w_{i,n}}$指向相似方向

Word2vec 优化

- 两套词向量，使用时只保留一套
- 优化方法: 随机梯度递降(SGD)
- 每次扫描到一个词，算一下梯度，更新……
- 收敛很快。大语料：1、2个pass，小语料：~10个pass

词嵌入算法的评估

- 词相似性任务
  - 哪一对更接近？

tiger　　cat

street　　children

- 类比任务
  - Beijing to China is like Berlin to ____
- 传统NLP任务
  - Named Entity Recognition (命名实体识别)， Noun phrase chunking (名词短语识别)

不同算法性能比较

	Similarity Tasks				Analogy Tasks		NLP Tasks	
Method	WS	WR	MEN	Turk	Google	MSR	NER	Chunk
word2vec	74.1	54.8	73.2	**68.0**	**72.3**	**63.0**	**84.8**	94.8
SVD	69.2	60.2	70.7	49.1	24.0	11.3	81.2	94.1
GloVe	75.9	63.0	75.6	64.1	54.4	43.5	84.5	94.6
Singular	76.3	**68.4**	74.7	58.1	50.8	39.9	83.8	94.8
Sparse	74.8	56.5	74.2	67.6	71.6	61.9	78.8	**94.9**
PSDVec	**79.2**	67.9	**76.4**	67.6	62.3	50.7	84.7	94.7

- NER: 命名实体识别，Chunk: 名词短语识别

词嵌入相关资源

- Word2vec, C语言实现
- Gensim，Python实现，接口多
  - >>trained_model.most_similar(positive=['woman', 'king'], negative=['man'])

    [('queen', 0.50882536), ...]
- DL4J, Deep Learning for Java
- 我自己的PSDVec(未来topicvec也会在这里)
- 分词软件 jieba, Python实现

语料资源

- 英文Wikipedia 12G
- 中文Wikipedia 1.2G
- http://lafnews.com/corpus/ 只含新闻链接(需自己使用爬虫抓取)
- 人民日报1998年1月词性标注语料
- 500万微博语料
- 分词词库
- 命名实体识别语料(英文)和训练代码
- 使用词嵌入的CRF Chunking代码和数据

谢谢大家！

欢迎大家批评指正！