09 Lexical Semantics

同义词 synonym
反义词 antonym

上义词 hypernym
下义词 hyponym

整体词 holonym
部分词 meronym

同音同形异义词 homonym
一词多义 polysemy

Model 只比较词汇的在字面上是否相同, 无法读懂词汇的意思, 在模型看来之间, movie-film 就是两个不同的词, 不会知道它们其实是是同一个东西. 而且对于一些从没在 train set 里面出现过的词 (out-of-vocabulary) 模型更是一脸懵逼. 所以我们需要告诉模型每个词的 “meaning”

Word Semantics

如何向模型表达一个词的意思呢? 我们有两种思路:

Lexical Semantics 词汇语义

Lexical semantics 研究词汇的meaning以及这些meaning之间的关系. 存在一些现成的 Manually constructed resources: lexicons, thesauri, ontologies, etc. 这些 database 储存了各个词汇的 meaning 之间的关系, 比如说同义, 反义, 上义, 下义.

注意: 词汇语义只关注各个词汇单独的 meaning, 不关注上下文.

Distributional Semantics 分布语义???

How words relate to each other in the text. Automatically created resources from corpora.

注意: 分布语义基于上下文来理解得出词汇意思

Lexical Semantics 词汇语义

一个 word 可以有多个 meaning, 一个 meaning 也可以用不同 word 来表达.

Meaning 之间的关系可以有以下几种:

- Synonyms (same) and antonyms (opposite/complementary)
- Hypernyms (generic), hyponyms (specific)
- Holoynms (whole) and meronyms (part)

WordNet

A database of lexical relations. (can be access via NLTK).

Synset

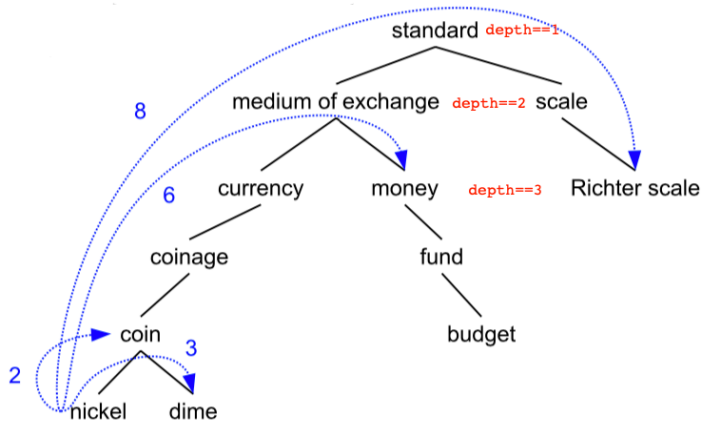
WordNet 有一种叫 synset 的东西. 一个 synset 代表一种 meaning, 每个synset 里面都包含很多能够表达这个 meaning 的 word (lemma). WordNet 里, lemma 的形式是 `[Lemma('brother.n.01.brother')]`, 是一个 synset+词形的组合

```
>>> from nltk.corpus import wordnet as wn
>>> wn.synsets('bank')← "bank"一词有哪些 meaning
[Synset('bank.n.01'), Synset('depository_financial_institution.n.01'),
Synset('bank.n.03'), Synset('bank.n.04'), Synset('bank.n.05'),
Synset('bank.n.06'), Synset('bank.n.07'), Synset('savings_bank.n.02'),
Synset('bank.n.09'), Synset('bank.n.10'), Synset('bank.v.01'),
Synset('bank.v.02'), Synset('bank.v.03'), Synset('bank.v.04'),
Synset('bank.v.05'), Synset('deposit.v.02'), Synset('bank.v.07'),
Synset('trust.v.01')]
>>> wn.synsets('bank')[0].definition()←第一个 meaning 的定义
u'sloping land (especially the slope beside a body of water)'
>>> wn.synsets('bank')[1].lemma_names() ←有哪些词可以表达第二个 meaning
[u'depository_financial_institution', u'bank', u'banking_concern',
u'banking_company']
```

Word Similarity (Based on Hypernym/Hyponym Tree)

Money 和 nickel 这两个 concept (meaning/synset) 是相关的, 但是它们在 WordNet 并没有一个直接相连的 relationship (同义反义上义下义), 那么我们应该如何描述 money 和 nickel 的相关关系呢? 没关系我们有hypernym/hyponym tree, 它可以帮助表达两个 concept 的相关度.

Hypernym/Hyponym Tree



Path Similarity

$simpath(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)}$ where c_1, c_2 是两个 concept (meaning/synset)

- simpath(nickel, money) = 1/6 = 0.17

存在问题:

上面几层词的差距其实非常大, 跳过一步可能意思已经离了已经十万八千里, 所以我们还需要利用 depth 信息

Wu & Palmer Similarity

引入最低公共母节点 lowest common subsumer (LCS) 概念.

$$\text{simwup}(c_1, c_2) = \frac{2 * \text{depth}(\text{LCS}(c_1, c_2))}{\text{depth}(c_1) + \text{depth}(c_2)}$$

- simwup (nickel, money) = $\frac{2*2}{3+6}$

Lin Similarity

- Given a corpus, $p(C)$ is the prob of a word to be an instance of concept C in the corpus: $p(C) = \frac{\sum_{word \in lemma(C)} count(word)}{N}$ where N 是整个语料库的单词数 (总 freq).
- Information content (IC) 描述了一个 concept 带来的信息量, 一个 concept 越少见, 所带来的信息量越大: $IC(C) = -logP(C)$
- Lin distance 描述两个 concept 的相似度: $simlin(c_1, c_2) = \frac{2 * IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)}$

Words Sense Disambiguation (WSD)

给 text 里面的每个 word 贴上一个 sense 标签来消除歧义. 如果消歧在很多 NLP task场景下非常有用, 但是现实中大家往往跳过这一步, 因为做一盒好的消歧太难了. 这是一个活跃的新研究领域.

Supervised Approach

Apply standard machine classifiers

Input: feature vector that is typically composed by context words syntax around the target word

Output: sense label of the target word

注意: 这个方法需要有一个预先准备好的sense-tagged corpus. eg. SENSEVAL, SEMCOR (available in NLTK). 要建立这样的语料库非常花时间!

Less-Supervised Approach

- **Lesk:** 在target word 的各个 meaning的定义中找找看有没有跟 target word 的上下文重合的词. 比如说 “bank” 一次有一个 “岸” 的意思, 这个意思的定义里面有”river”, “lake”, “sloping”, “water” 等词. 如果我们的 target word “bank”的上下文中也有这些词, 说明这里的 “bank” 很可能就是岸的意思. 如果我们的 target word “bank”的上下文中提到了 “money”, “currency”, “deposit”, 那么这个 “bank” 很可能就是银行的意思.

10 Distributional Semantics

Distributional Semantics

基本思想: a word is characterized by the company it keeps. 词的意思可以通过它的上下文来猜

Distributional hypothesis: linguistic items with similar distributions have similar meanings. 如果两个 word 经常出现在相同的上下文中, 说明它们的意思很相似.

Distributed Semantics

基本思想: 用数字 vector 来表达一个词的意思 (而不是用字面词形来表达), 两个 vector 的 distance 很近说明这两个词的意思很相似.

在 information retrieval 中, 我们用一个 document 中的 words 来把这个 document 写成 vector, 这个 vector 的长度就是 vocabulary size, 如果两个 document/query vector 的夹角很小, 说明他们很相似. 现在, 我们要想办法把 word 写成 vector, 如果两个 word vector 很接近, 说明他们很可能是同义词. 这也是一种 vector space model.

Dimensionality Reduction

Term-document matrices are very sparse. We want to reduce its dimensionality and make the matrix more compact, make the vectors shorter and denser.

SVD

To find a lower-rank (k-rank) approximation of the original term-document matrix. Choose the k value such that the sum of squared error between the original matrix and the approximation matrix reached its minimum.

Word-Word Matrix

基本思想: 两个 word 越经常一起出现, 就越 related (related不一定就是同义词了, 也可能是反义词, 比如 heaven and hell)

先做很多个 context window, 一个 word pair 中的两个 word 在多少个 context window 中同时出现了, 把这个 count 填到 cell 里面. 对于 vocabulary 里面的所有可能 word pair 都做一样的事, 最后形成的 matrix size 是 $|V| \times |V|$:

	…	the	country	hell	…	Σ
…						
state		1973	10	1		12786
fun		54	2	0		633
heaven		55	1	3		627
…						
Σ		1047519	3617	780		15871304

在几个 context window, 中 the & state 一起出现了

直接填 count 会存在一个问题: 常见词, 比如 “the” 的计数肯定会很大, 别的 pair 有 count 100 次已经了不得了, 包含 “the” 的 pair count 1000 次还不算个事. 所以我们找到了 PMI 来代替 raw count.

Pointwise Mutual Information (PMI)

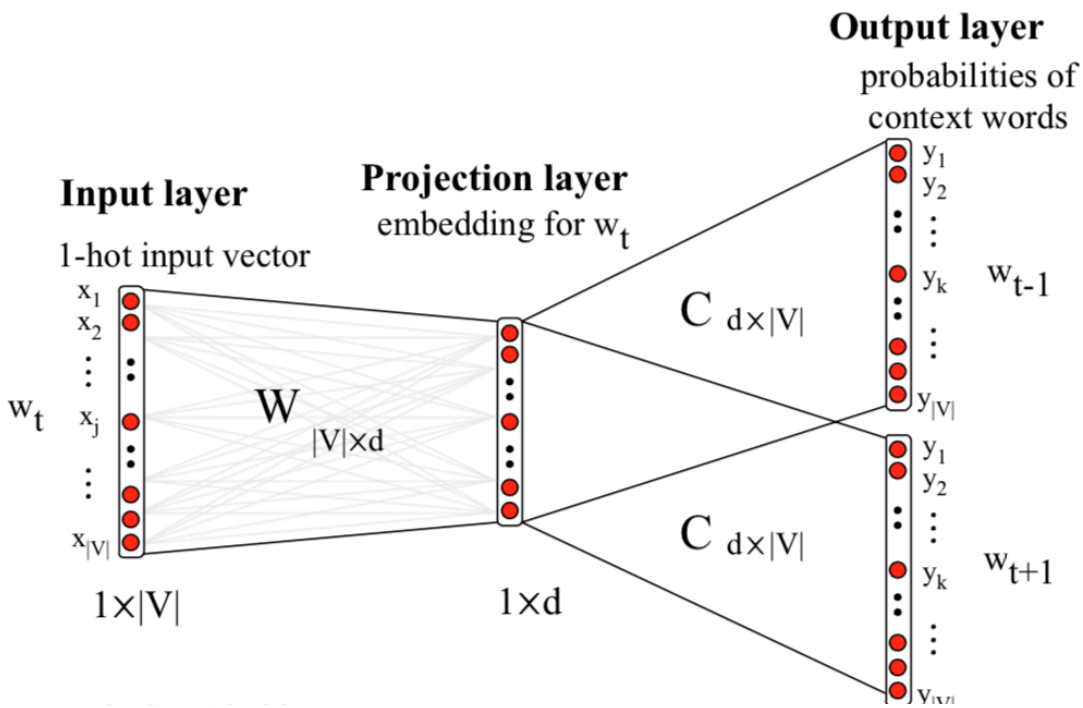
$$PMI(w_1, w_2) = log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)}$$

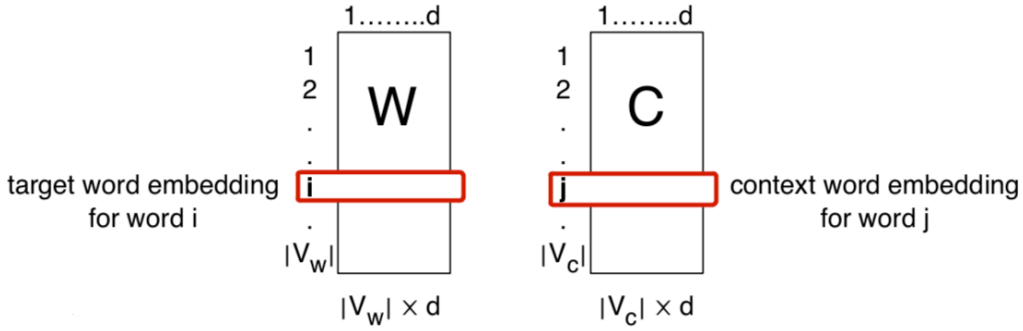
如果两个词完全从未一起出现, PMI 会是负无穷. 若果两个词的出现完全independent, PMI 会是 0. 如果两个词的出现有共同性, PMI 会是个正数, PMI 数值越大, 表示这个两个词越经常一起出现. PMI 使用了 P(the)这样的 term 作为除数, 所以common words dominate the matrix 的问题解决了. 负数 PMI 一般不提供信息, 所以我们可以把所有负数 PMI 都变为 0.

Word Embedding (Also Reduce Dimension)

一种用 shorter and denser vector 表达 word 的方法.

- Word vector 一般有300~1000 dimension
- Capture semantic similarity between words (eg. movie vs. film)





基本思想: Learn words vector representation such that words with similar context have similar spatial positions.

Google 的 word2vector 包里面有两种模型:

CBOW (Continuous Bag of Word)

Given context, predict target

Skip-Gram

Given target, predict context.

- Prob of a context window given a target word can be written as: $\prod_{l \in -L,\ldots,-1,1,\ldots,L} P(w_{t+l}|w_t)$

Training a skip-gram model is a process to find the 2 parameter-matrixes $\mathbf{W}$ and $\mathbf{C}$ that maximum the likelihood of raw text.

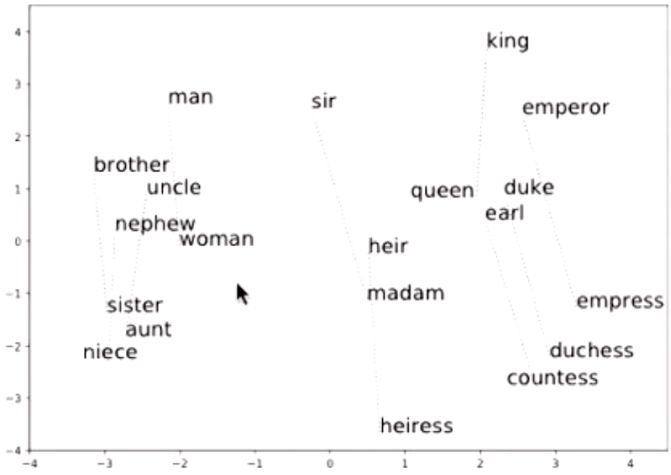
Evaluation Word Embedding

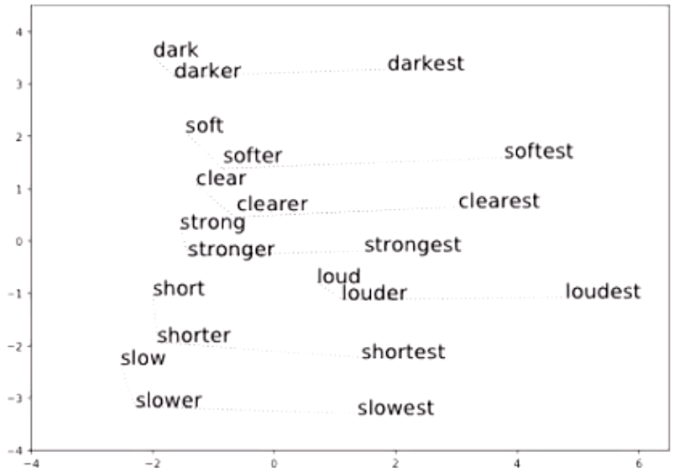
Lexicon Style Tasks

找一个人工编撰的辞典, 辞典里面对于各个 word-pair 都带有一个相关性数值. (就像 homework 2 那个golden standard 一样), 计算这些 word-pair 的 word embedding 的 cosine similarity, 看看它们是否和词典里的相关性一致 (相关性大的离得近, 相关性小的离得远). 如果是这样, 说明我们的 word embedding 很成功地表示了 word, 反之则不是 word 的正确表达.

Word Analogy Task 单词类比任务

man(vector) - king(vector) = woman(vector) - queen(vector) ?
darker(vector) - darker(vector) = soft(vector) - softer(vector) ?
如果像下图这样, 说明我们的 word embedding 很成功地表达了 word 的意思.


king
man
sir
emperor
brother
uncle
queen
duke
earl
nephew
woman
heir
madam
empress
sister
aunt
niece
duchess
countess
heiress


dark
darker
darkest
soft
softer
softest
clear
clearer
clearest
strong
stronger
strongest
loud
short
louder
loudest
shorter
slow
shortest
slower
slowest