词向量评估

词向量的评价方式分为内部评价和外部评价。

内部评价是我们在词向量训练技术中的具体的中间子任务上的评价。比如类比，这些子任务通常简单快速，能够帮助我们理解词向量，对于词向量的性能有具体的数值指标。这样做主要是因为一个具体的机器学习任务通常需要很多的时间和计算资源来完成，且词向量的性质会受到机器学习模型的好坏的干扰，所以有时候我们没有必要涵盖下游具体的任务去评估。

外部评价就是在真实任务上的评价，词向量的评价依赖于任务的结果，毕竟我们训练词向量是为了完成具体任务，所以这也是必不可少的，并且很多时候我们需要根据具体的任务来选择合适的词向量训练方法。然而当某个具体任务的模型表现很差时，我们常常并不能弄清楚到底是那部分有问题，所以还需要内部评价的支撑。

# 内部评估

内部评价指的是用词的词性，相关性等内部固有关系来评价生成的词向量的好坏。 外部评价指的是用生成的词向量去作为下游任务的输入，看哪种词向量可 以更好的实现下游任务。内在评估又可以分为绝对内部评价（absolute intrinsic evaluation）和相对的内部评价（comparative intrinsic evaluation）。

## 绝对内在评估

绝对内在评估直接衡量给定两个单词之间的句法和语义关系。共有四种类型的评价：

* 相关性（Relatedness）：比较生成的词向量的词与词之间的余弦相似度和人类评价的相似度的关系
* 类比性（Analogy）：对于一个y，去找到一个x，使得x:y的关系要和a:b的关系一样
* 分类（Categorization）：把生成的词向量做聚类，看聚类是否准确
* 选择偏好（Selectional preference）：确定一个词是某个动词的主语还是宾语

### 相关性(Relatedness)

对于两个单词，他们之间的余弦相似度应该和人类主观评价的得分有较高的相关性。即评价词向量模型在两个词之间的语义相关性，如：学生与作业，中国与北京等。具体方法由监督模式实现，首先需要一份如下的标记文件，一般可以由人工标注：

学生 上课 0.78

教师 备课 0.8

...

上述文件代表了词语之间的语义相关性，我们利用标注文件与训练出来的词向量相似度进行比较，如：词向量之间的cos距离等，确定损失函数，便可以得到一个评价指标。

但这种方法首先需要人力标注，且标注的准确性对评价指标影响非常大。

### 类比性(Analogy)

假设给了一对单词(a, b)和一个单词c，task会找到一个单词d，使得c与d之间的关系相似于a与b之间的关系：



比如：queen-king+man=women

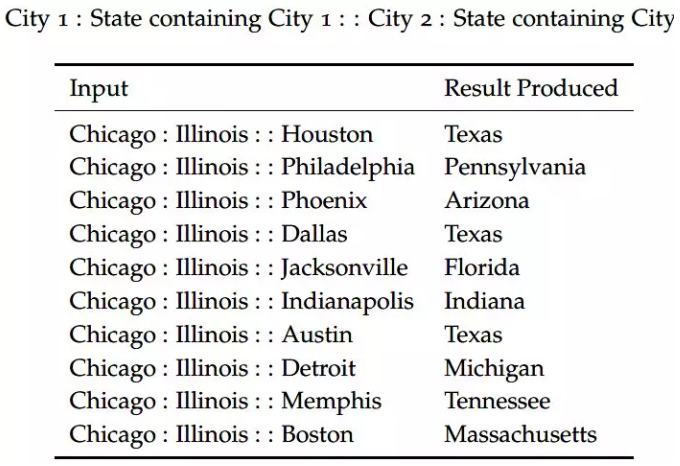
在给定word embedding的前提下，一般是通过在词向量空间寻找离(b-a+c)最近的词向量来找到d。

也就是，哪个词与c的关系跟a与b的关系很像？具体的数学表达为：



这个式子的来源是xb-xa=xd-xc，亦即xd=xb-xa+xc。

在使用类比评价时要非常小心，类比条件的选取要仔细考虑。看一下下面这个例子：

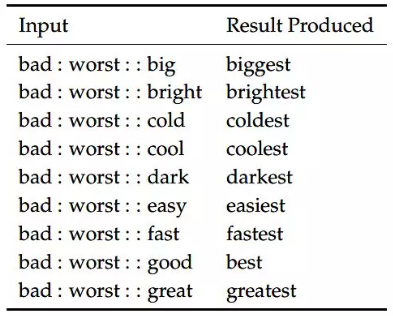


这是同一个国家的两个城市的类比，我们发现类比的结果比较糟糕。原因很明显，我们选取的这个类比问题的答案太多了！甚至于还有很多同名的美国城市、城镇或者乡村。如果是下面这个类比问题：

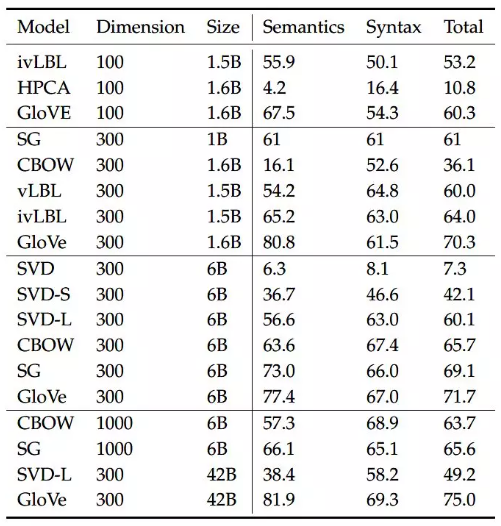


你需要注意的是你的数据集的时效，因为首都城市可能会有变化。

我们不仅可以在语义上做类比，还能在语法上做类比：



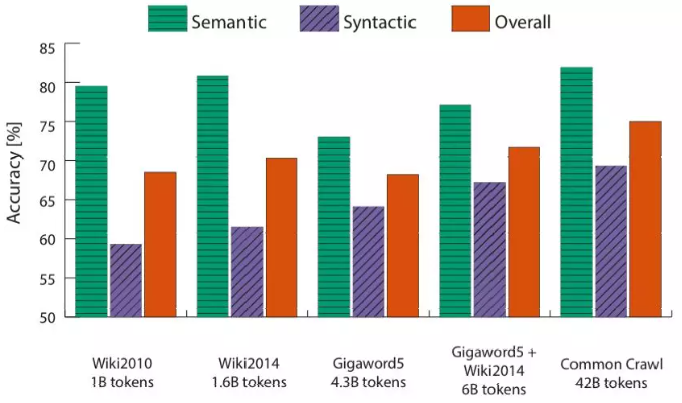
下面我们在不同大小的数据集上做不同模型不同向量维度的评估：



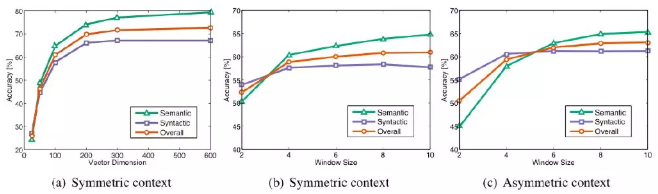
观测到三个主要的结论：

* 词向量的性能非常依赖训练模型的选取，毕竟不同的模型能利用上的数据信息都不一样。这一点并不意外，因为这些生成词向量的方法所依据的特性是完全不同的(如同时出现的次数，奇异向量等。)
* 数据集越大，性能越好，这一点和机器学习的普遍规律是相同的。这是因为，例子越多，生成的系统学习到的经验就更丰富。比如在完成词汇类比的例子中，系统如果之前没有接触测试词，就可能会生成错误的结果。
* 向量维度过大或过小都会导致性能的下降，这一点可以从机器学习普遍规律推测得到，维度过小就不能很好地捕获数据集上的各种信息，也就是高偏差，underfitting；维度过大，就会捕获过多的数据上的噪声，泛化能力变差，也就是高方差，overfiiting。

下图表现了数据集对性能的影响：



下图表现了向量维度和窗口大小的影响：



除类比法之外，我们还可以使用人工标注的词性关系数据来验证我们的词向量的性能。这种方式人力成本很高，但是可操控性强。

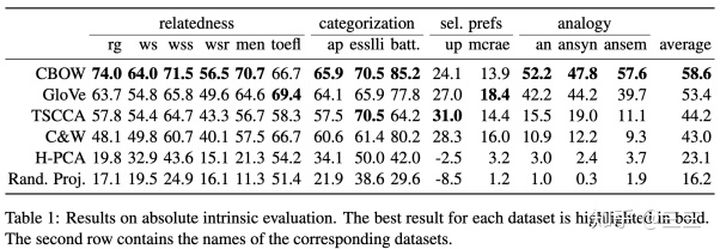
### 分类(Categorization)

把词聚类成不同的堆，看是否聚类准确。

### 选择偏好(Selected preference)

判断某名词是更倾向做某个动词的主语还是宾语，例如一般顺序是he runs而不是runs he。

评价结果如下,可以看出，绝大多数任务中，CBOW表现最好。但是个别任务里，其他词向量更好。

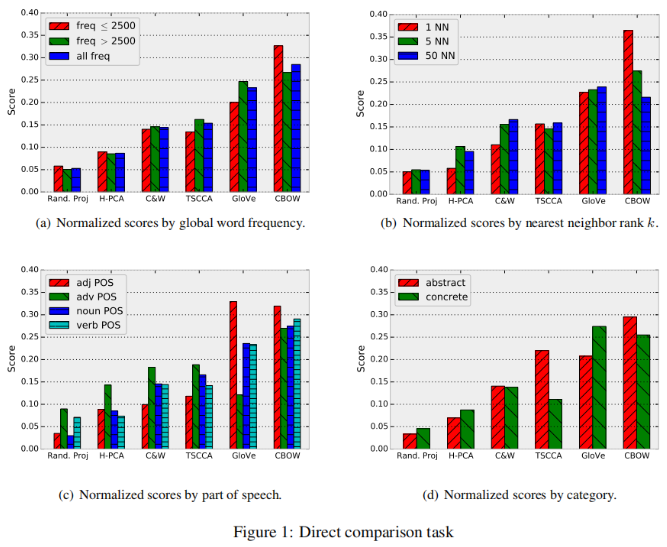


## 比较内在评估 comparative intrinsic evaluation

在相对内部评价中，用户直接来判断词向量的好坏。作者的具体做法如下：

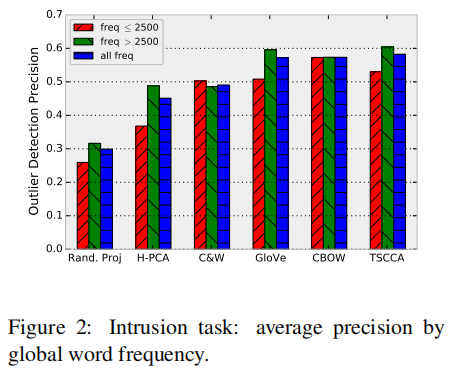
* 选取了词频，词性和词义不同的100个单词（选择10种类别的词，每种类别里有一个形容词，一个动词，4个名词，4个动词）
* 找出每个词的n nearest neighbors, 选取rank为1，5，50的neighbor。所以对于6种词向量，对于每一个词，我们分为计算出rank为1，5，50的neighbor。
* 让人类来分别评价6中词向量中，rank 1，5，50的neighbor里哪个与选定词最近。

结果如下，同样可以看出，没有一种词向量是在所有任务中都表现最好的



在相似度（relatedness）的比较中，我们对于任意一个单词，我们只找了一个相近的单词，这并不理想（因为每个单词都有很多近义词）。所以作者提出了一种新的衡量方式：Coherence。对于每一个单词，事先选出两个近义词和一个不相关的词，看用生成的词向量能否辨别出不相关的词。

结果如下，可以看出不同词向量的生成方法，对于不同词频的单词，所得到的结果是不同的。



## 内部评估的特点

* 一般是在一个特定的子任务中进行评测
* 评估速度快，易于计算
* 有助于理解相关的系统
* 需要与实际任务建立关联，才知道得到的词向量是否有用

# 外部评估

## 外部评估理论

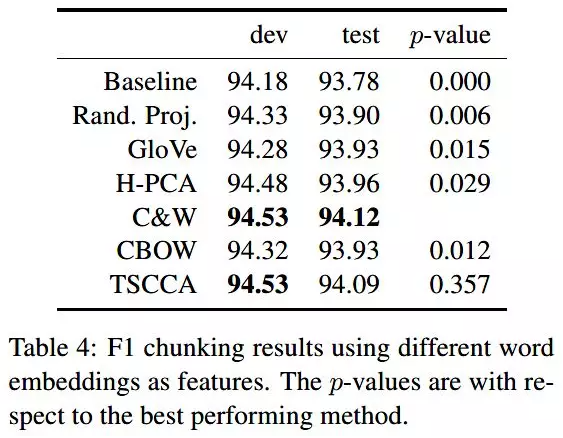
外部评估衡量的是词向量模型对一个具体任务的影响。在外部评估中有一个隐含的假设，那就是对词向量的质量有一个一致的、全局的排名，高质量的词向量必然会改善下游任务的性能。但是，这个和假设并不成立：不同的任务需要不同的词向量。所以外部评估仅仅用来比较词向量之间的相对优劣势，而不能当作词向量质量的广泛意义上的评估。

使用此类评估存在一个隐含的假设，即单词嵌入质量是有固定排名的。也就是说，嵌入模型无论在什么任务里的表现排名应该是基本一致的。因此，更高质量的嵌入将必定会改善任何下游任务的结果。

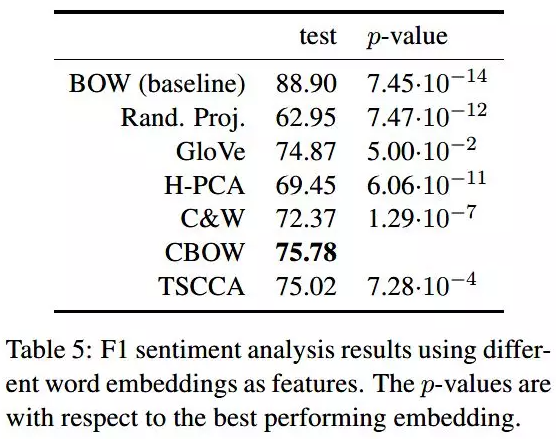
本文选取了以下两种下游任务来评判：

* Noun phrase chunking：名词分块
* Sentiment classification：情感分类

名词短语分块任务，注重语法上的分类，结果如下：



情感分析任务，注重语义分析，结果如下：



没有一种词向量可以在所有下游任务中都表现最好，最优的处理方式就是根据具体的任务选取最优的词向量模型，所以对于不同下游任务，我们应该尝试不同词向量的表示。

## 外部评价的特点

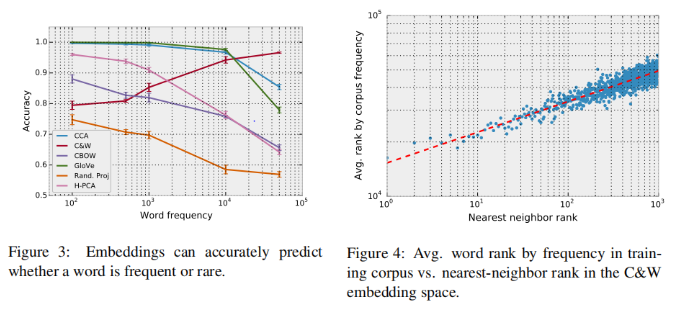
* 在一个实际任务中进行评测
* 需要花很长的时间来才能得到精度
* 有时无法确定具体是什么原因导致任务表现出现差异，因此难以合理地对词向量进行评估
* 如果替换原有的子系统后获得精度提升，则说明替换很可能是有效的

# 词频信息

最后，作者通过以下两种实验发现了词向量里面包含词频信息。

* 用词向量来预测单词在语料中词频
* 对于所有在WordSim-353数据集的单词，研究其K=1000 nearest neighbors和他们在语料中词频的大小排序。

结果如下，可以看出，我们可以通过词向量来较好的预测单词的词频，其中GloVe和CCA中包含了较多的词频信息。另外单词的词频与其在语料库里的词频排名也有很强的相关性



# 词向量再训练

多数NLP任务都可以认定为分类任务，典型的有情感分析和命名实体判别。对于分类任务，我们设想这样的一个数据集：



x为d维词向量，y是c维的one-hot向量标注。

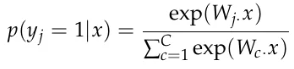
在其他机器学习任务中，我们都是给定输出和标注，训练模型的参数，但是，在NLP任务中，我们还可以有这样一种训练思路：对输入词向量进行再训练。

在经过内部评估后，我们选取了比较合适的训练策略得到了较好的词向量，在很多情况下，这样的词向量已经可以在外部任务中取得很好的效果，但是，也有可能在外部任务中再训练以获得更好的性能。不过，再训练词向量是有风险的。

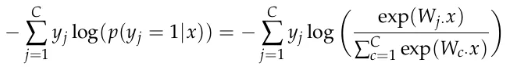
如果想要再训练词向量，首先要保证数据集足够大，能覆盖到词库中的大部分词，这是因为我们使用word2vec或者glove训练词向量时，所有的词向量都在同一个词空间中，而如果在一个较小的数据集上再训练，那么使得经过再训练的词向量发生了空间的偏移，导致没有训练的词向量与之存在于不同的词空间，这样自然会使得性能下降。所以，小数据集上并不能使用再训练。

接下来让我们考虑一个softmax分类任务。

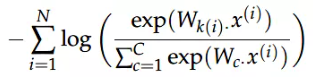
使用softmax函数计算分类概率：



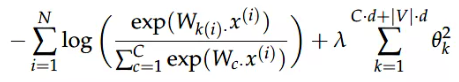
使用cross-entropy函数计算损失函数：



假设k(i)是样本i中one-hot向量y中值为1的维度的索引，那么在N个样本点上的损失函数为：

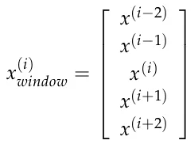


如果我们要进行再训练，那么在训练过程中我们更新的参数会非常多，也就非常容易过拟合，所以需要添加一个正则化项：

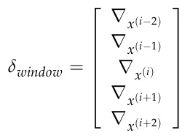


在实际任务中，使用单个的词向量进行分类是很困难的，因为人类语言的单词往往蕴含着多重含义，需要根据上下文进行语义判别，因此，模型的输入一般是词向量的序列，这个序列包括中心词和它的上下文单词。上下文的单词数称为窗口大小，其选择取决于任务的性质，一般语法任务会选取较小的窗口，语义任务会选取较大的窗口。

在上面的sotfmax分类任务中，我们可以把输入词向量变成：



同时，在训练中更新词向量时，也会更新整个窗口中的词向量：



# 实践

这里给出部分词向量评估的代码，请查看github链接：

<https://github.com/jpegbert/NLP_Coding/tree/master/english_chinese_embedding_evaluation>

# 总结

词向量的评估没有标准的评估方法，需要针对具体问题进行合理的思考也探索，但是针对部分问题还是可以合理利用本文提到的方法进行解决。

# 参考：

<https://mp.weixin.qq.com/s/4sqMV3tS146VNrTyyncYnQ>

<https://mp.weixin.qq.com/s/RrR-iAdlnuSnRai1cwcw3A>

<https://mp.weixin.qq.com/s/5KQoya9BjmCOJAD9_GaynA>

<https://mp.weixin.qq.com/s/hbQrdzgDbibCI6o2WzhR-A>

<https://zhuanlan.zhihu.com/p/156828242>

<https://xubujie.github.io/DataScienceBlog/nlp/2020/07/04/Evaluation-methods-for-unsupervised-word-embeddings.html>