这篇文章主要讲了Word2Vec能够成功的一些因素，会和传统的分布式语义模型(DSMs)进行比较，并将一些技巧迁移到分布式模型上去，会显示出DSMs并不会比word2vec差，虽然这并不是新的见解，但由于这些传统方法在深度学习面前总是显得黯然失色，因此值得被再提及。这篇博客主要基于*Improving Distributional Similarity with Lessons Learned from Word Embeddings* 。

Glove模型

GloVe明确的做了SGNS(skip-gram with negative-sample)所含蓄表达的思想：将embedding 空间的**向量偏移**编码成语义信息。这在word2vec中是一个偶然的副产品，但却是Glove的目标。

> *Encoding meaning as vector offsets in an embedding space -- seemingly only a serendipitous by-product of word2vec -- is the specified goal of GloVe.*
>
> *If we dive into the deduction procedure of the equations in GloVe, we will find the difference inherent in the intuition. GloVe observes that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. Take the example from StanfordNLP (Global Vectors for Word Representation), to consider the co-occurrence probabilities for target words **ice** and **steam** with various probe words from the vocabulary:*
>
Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96
>
> - *“As one might expect, **ice** co-occurs more frequently with **solid** than it does with **gas**, whereas **steam** co-occurs more frequently with **gas** than it does with **solid**.*
> - *Both words co-occur with their shared property **water** frequently, and both co-occur with the unrelated word **fashion** infrequently.*
> - *Only in the ratio of probabilities does noise from non-discriminative words like **water** and **fashion** cancel out, so that large values (much greater than 1) correlate well with properties specific to **ice**, and small values (much less than 1) correlate well with properties specific of **steam**. ”In this way, the ratio of probabilities encodes some crude form of meaning(粗糙的语义信息) associated with the abstract concept of thermodynamic phase. //第三行那个比例就包含的语义信息，比如k=solid的时候，$p(k|ice)/p(k|stream)$的值很大，说明ice更多的是和solid相关的，而stream则不太跟固体相关，这里就包含了一些热力学里面的语义信息。*

> *However, Word2Vec works on the pure co-occurrence probabilities so that the probability that the words surrounding the target word to be the context is maximized..*

1. 目标方程

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

这个目标方程所要表达的是：两个单词的共现概率的比率（而不是它们的共现概率本身）是包含信息的，并且旨在将该信息编码为**向量差异**。为了达到这个目的，他们提出了加权最小二乘目标函数，旨在最小化两个单词的向量的点积与它们的共现数的对数之间的差异。其中，w_i, b_i是单词i的词向量和偏差，$\tilde{w_j}, \tilde{b_j}$是关于单词j的词向量和偏差；$X_{i,j}$是单词i和j的共现数量，f是一个权重函数，给共现数量较少的单词对给予更小的权重。

共现关系可以用一个矩阵表示，那么就不需要用整个语料库作为输入了。

2. word2vec与Glove的区别

1. GloVe与word2vec，两个模型都可以根据词汇的“共现co-occurrence”信息，将词汇编码成一个向量（所谓共现，即语料中词汇一块出现的频率）。两者最直观的区别在于，word2vec是“predictive”的模型，而GloVe是“count-based”的模型，但是Glove又结合了机器学习的思想，是一种traditional和机器学习相结合的模型。

   > *To better explain this question, I'd like to include LDA for comparison. Before GloVe, the algorithms of word representations can be divided into two main streams, the statistic-based (**LDA**) and learning-based (**Word2Vec**). LDA produces the low dimensional word vectors by singular value decomposition (SVD) on the co-occurrence matrix, while Word2Vec employs a three-layer neural network to do the center-context word pair classification task where word vectors are just the by-product.*
   >
   > *The most amazing point from Word2Vec is that similar words are located together in the vector space and arithmetic operations on word vectors can pose semantic or syntactic relationships, e.g., "king" - "man" + "woman" -> "queen" or "better" - "good" + "bad" -> "worse". However, LDA cannot maintain such linear relationship in vector space.*
   >
   > *The motivation of GloVe is to force the model to learn such **linear relationship** based on the co-occurreence matrix explicitly. Essentially, GloVe is a log-bilinear model with a weighted least-squares objective. Obviously, it is a hybrid method that uses machine learning based on the statistic matrix, and this is the general difference between GloVe and*

> *Word2Vec.*

- Predictive的模型：如Word2vec，根据context预测中间的词汇，要么根据中间的词汇预测context，分别对应了word2vec的两种训练方式cbow和skip-gram。对于word2vec，采用三层神经网络，然后基于hierarchical softmax或者sample-base softmax进行训练。
- Count-based模型：如GloVe，本质上是对共现矩阵进行降维。首先，构建一个词汇的共现矩阵，每一行是一个word，每一列是context。共现矩阵就是计算每个word在每个context出现的频率。由于context是多种词汇的组合，其维度非常大，我们希望像network embedding一样，在context的维度上降维，学习word的低维表示。这一过程可以视为共现矩阵的重构问题，即reconstruction loss。(这里再插一句，降维或者重构的本质是什么？我们选择留下某个维度和丢掉某个维度的标准是什么？Find the lower-dimensional representations which can explain most of the variance in the high-dimensional data，这其实也是PCA的原理)

2. 两种方法都能学习词的向量表示，两者的performance上差别不大，两个模型在**并行化**上有一些不同，即GloVe更容易并行化，所以对于较大的训练数据，GloVe更快。
3. 两者所得到的结果有所不同。

> *In the practice, to speed up the training process, Word2Vec employs negative sampling to substitute the softmax fucntion by the sigmoid function operating on the real data and noise data. This emplicitly results in the clustering of words into a cone in the vector space while GloVe's word vectors are located more discretely.*

4. 在英文上，glove for GloVe 和 gensim for word2vec是常用的训练词向量的python package，完全可以使用自己的训练语料训练词向量。当然，他们都提供了google news（英文）上训练好的词向量，大家完全可以下载下来，直接使用。对于中文的训练语料，可以使用sogou中文新闻语料。

Neural Word Embedding vs.DSMs

word2vec的成功，很大的原因在于他是基于学习的predictive模型，他进行预测周围的单词，这相对于基于同现计数的模型更有天然的优势。2014年，Baroni等表明预测模型几乎在所有任务上面打败了计数模型。但是，我们从Glove来看，差异似乎并不那么明显，Glove对一个单词同现矩阵进行分解，这种思想类似于传统的PCA,LSA等方法，而Levy等[4]表明word2vec其实是潜在的分解了一个单词上下文的PMI矩阵。所以，从根本上来看，两类方法其实都是在针对数据中单词同现的数据进行分析。

为什么都是用的同样信息，而Word Embedding效果更好?

1. Models

这些模型用来作比较，以分析哪些方面能被应用到DSM中去。

- PPMI(**Singular Value Decomposition**)
  - PMI is a common measure for the strength of association between two words. It is defined as the log ratio between the joint probability of two words w and c and the product of their marginal probabilities: $PMI(w,c) = \log \frac{P(w,c)}{P(w)P(c)}$
  - *positive* PMI (PPMI):裁剪负值得到$PPMI(w,c) = \max(PMI(w,c), 0)$
- SVD(**Singular Value Decomposition**)
- SGNS(**Skip-gram with Negative Sampling**)
- GloVe(**Global Vectors**)

2. Hyper Parameters

以下数据处理等方面被用来比较，看是否可以应用到DSM中。

- 预处理
  - 动态的上下文窗口。在DSM中，使用静态窗口，而且没有被为单词加权；而SGNS和GloVe中，都对更近的词使用更大的权重，在SGNS中是通过冬动态上下文窗口实现的，在训练的时候通过在1和最大窗口之间进行均匀采样得到。
  - 数据下采样：对于出现频率太高的单词进行下采样，随机丢弃一些单词，如果这个单词以$p = 1 - \sqrt{\frac{t}{f}}$的概率大于阈值t的话。
  - 删除稀少的单词
- Association Metric
  - PMI作为一个单词相似性的度量标准，SGNS就是在潜在的分解一个PMI矩阵，基于此，衍生出来两个变种：
    - Shifted PMI：由于SGNS中负采样的个数影响PMI矩阵,产生 $log(k)$的偏移量,因此就有：

      $SPPMI(w,c) = \max(PMI(w,c) - \log k, 0)$
    - Context distribution smoothing：在SGNS中，通过smoothed unigram distribution采样负样本，也就是说对unigram分布加入指数α,经验性的设置为$\frac{3}{4}$，这就会使出现频率高的词更少概率的被采样到。然后就引入下面的PMI变种：

      $PMI(w,c) = \log \frac{p(w,c)}{p(w)p_\alpha(c)}$,且$p_\alpha(c) = \frac{f(c)^\alpha}{\sum_c f(c)^\alpha}$,f(w)表示单词w的频率。
- Post-processing

以下三个方面可以用来修改一个模型产生的word vectors。

- 添加上下文向量：GloVe中提出在word embedding上添加上下文向量，$\vec{v}_{\text{cat}} = \vec{w}_{\text{cat}} + \vec{c}_{\text{cat}}$,可以应用到PMI上；
- 特征值加权：SVD分解产生下面的结果: $W^{SVD} = U_d \cdot \Sigma_d$and $C^{SVD} = V_d$. 但是这些句矩阵有不同的属性: C^{SVD} 是标准正交化的, 但是 W^{SVD} 不是.相反，SGNS是更加对称的. 因此我们可以给矩阵 Σ_d 加权，通过使用额外的参数 p, 这个参数可以被调节:$W^{SVD} = U_d \cdot \Sigma_d^p$;
- 向量归一化： 将所有向量都归一化到单位长度。

揭穿先前的主张

Equipped with these insights, we can now debunk some generally held claims:

1. Are embeddings superior to distributional methods?
   With the right hyperparameters, no approach has a consistent advantage over another.
2. Is GloVe superior to SGNS?
   SGNS outperforms GloVe on all tasks.
3. Is CBOW a good word2vec configuration?
   CBOW does not outperform SGNS on any task.

一些建议

Finally -- and one of the things I like most about the paper -- we can give concrete practical recommendations:

- **DON'T** use shifted PPMI with SVD.
- **DON'T** use SVD "correctly", i.e. without eigenvector weighting (performance drops 15 points compared to with eigenvalue weighting with p=0.5p=0.5).
- **DO** use PPMI and SVD with short contexts (window size of 22).
- **DO** use many negative samples with SGNS.
- **DO** always use context distribution smoothing (raise unigram distribution to the power of α=0.75α=0.75) for all methods.
- **DO** use SGNS as a baseline (robust, fast and cheap to train).
- **DO** try adding context vectors in SGNS and GloVe.

Refference

注：此博客主要翻译了[7]，参考了[8]。

1. Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the Association for Computational Linguistics, 3, 211–225. Retrieved from https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570 ↩
2. Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical

Methods in Natural Language Processing, 1532–1543. http://doi.org/10.3115/v1/D14-1162 ↩
3. Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. ACL, 238–247. http://doi.org/10.3115/v1/P14-1023 ↩
4. Levy, O., & Goldberg, Y. (2014). Neural Word Embedding as Implicit Matrix Factorization. Advances in Neural Information Processing Systems (NIPS), 2177–2185. Retrieved from http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization ↩
5. Hamilton, W. L., Clark, K., Leskovec, J., & Jurafsky, D. (2016). Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Retrieved from http://arxiv.org/abs/1606.02820 ↩
6. Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. arXiv Preprint arXiv:1605.09096. ↩
7. http://ruder.io/secret-word2vec/index.html
8. https://zhuanlan.zhihu.com/p/31023929
9. https://www.quora.com/How-is-GloVe-different-from-word2vec