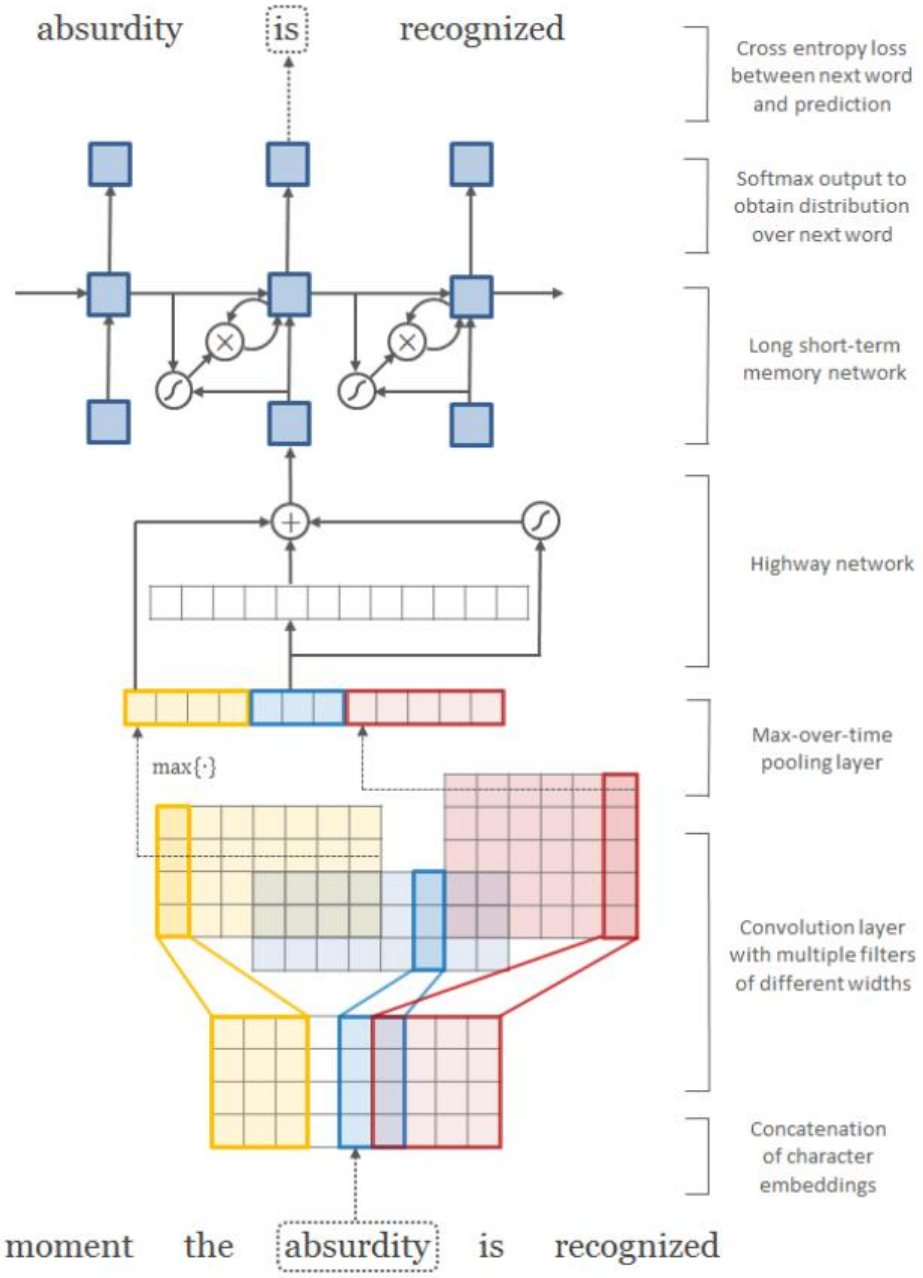
NLP项目
词向量二

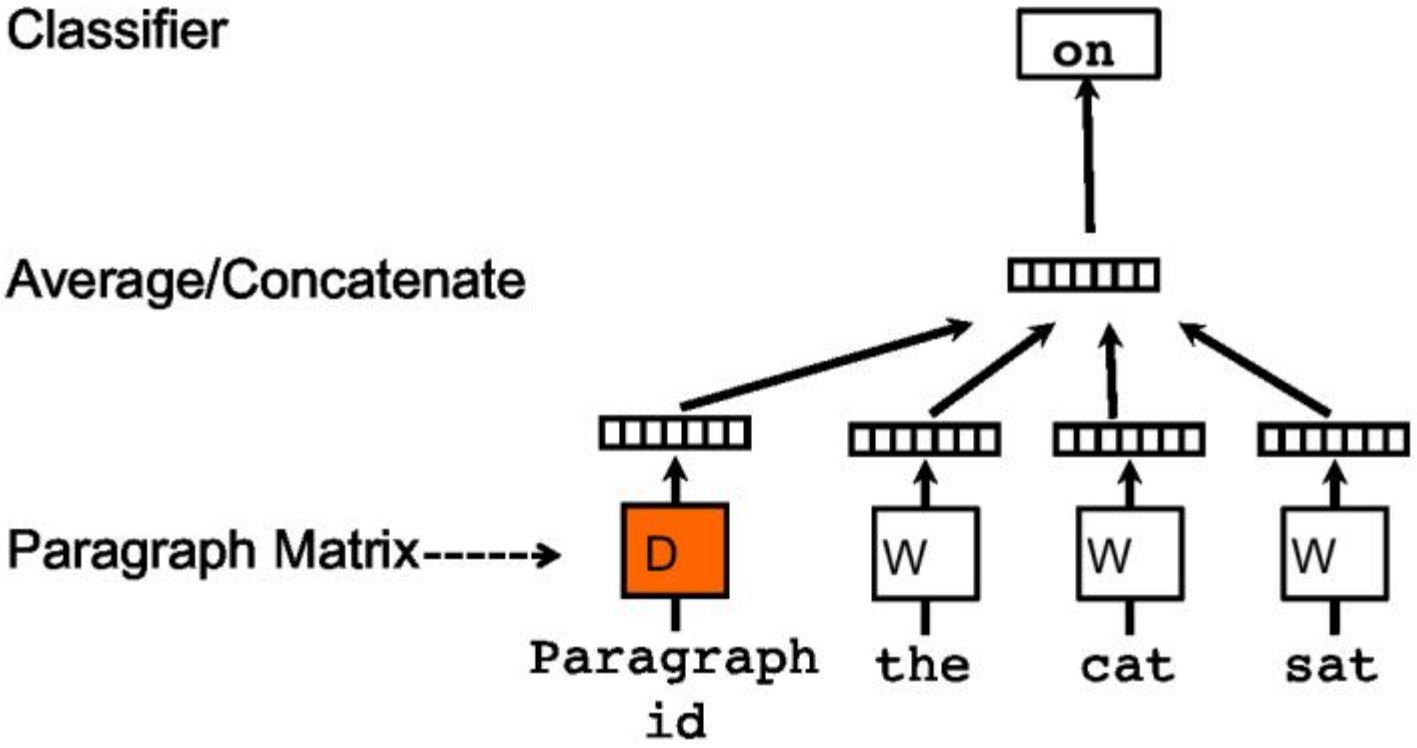
NLP基础_词向量_Char2Vec

- 和Word2Vec不一样，不进行分词，直接将字符转换为字向量，后续的模型基于字向量进行模型的构建。
- 相比于Word2Vec来讲，Char2Vec直接应用于字符集，对拼写更加宽容。现阶段NLP中应用非常多的一种向量方式。



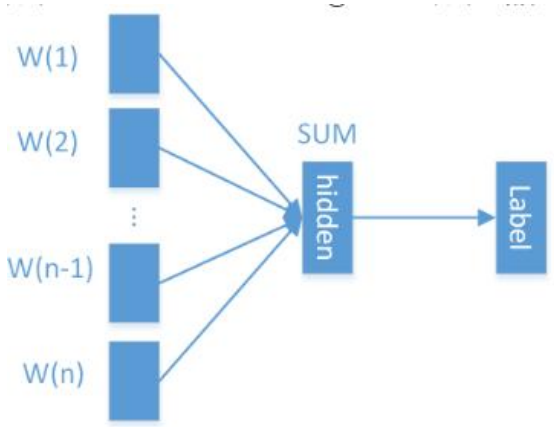
NLP基础_词向量_Doc2Vec

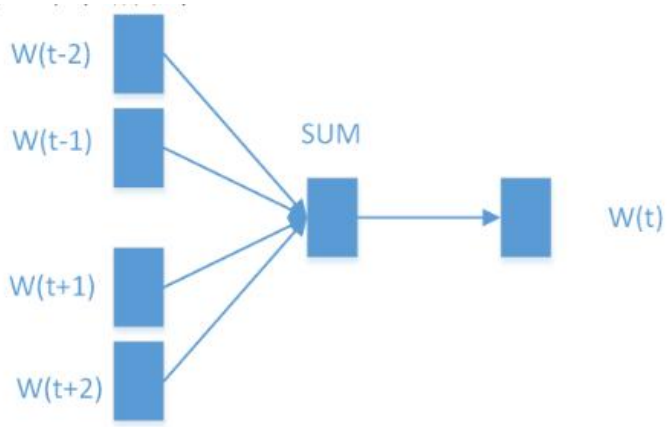
- Doc2Vec使用Word2Vec作为第一步输入，然后利用Word2Vec的单词向量对每个句子或者段落生成复合向量。



NLP基础_词向量_FastText

- FastText是一个文本分类和词向量训练工具/网络结构，最大的特点就是模型简单，只有一层隐层和输出层；结构基本的CBOW类似，主要区别在于：
  - **CBOW结构中预测的是中心词，FastText输出的是类别label；**
  - CBOW中输入是当前窗口除中心词之外的所有词，而FastText输入的是文章中的所有词；





NLP基础_词向量_FastText

- 在词向量的训练过程中，增加了subwords特性，其实就是一个词的character-level的n-gram，比如单词”hello”，长度至少为3的character-level的ngram的'hel', 'ell', 'llo', 'hell', 'ello'以及'hello'，每个ngram都可以使用一个dense的向量zg表示，故最终一个单词可以表示为:

$$V_{hello} = \sum_{g \in \phi} z_g^T v_c$$

NLP基础_词向量_FastText

$$h = \frac{1}{n}\sum_{i=1}^{n} w_i$$

$$z = sigmoid(W_o h)$$

$$loss = -\frac{1}{M}\sum_{i=1}^{m}\left(log\sigma(u_o^T h_i) + \sum_{j\sim P(w)}[log\sigma(-u_j^T h_i)]\right)$$

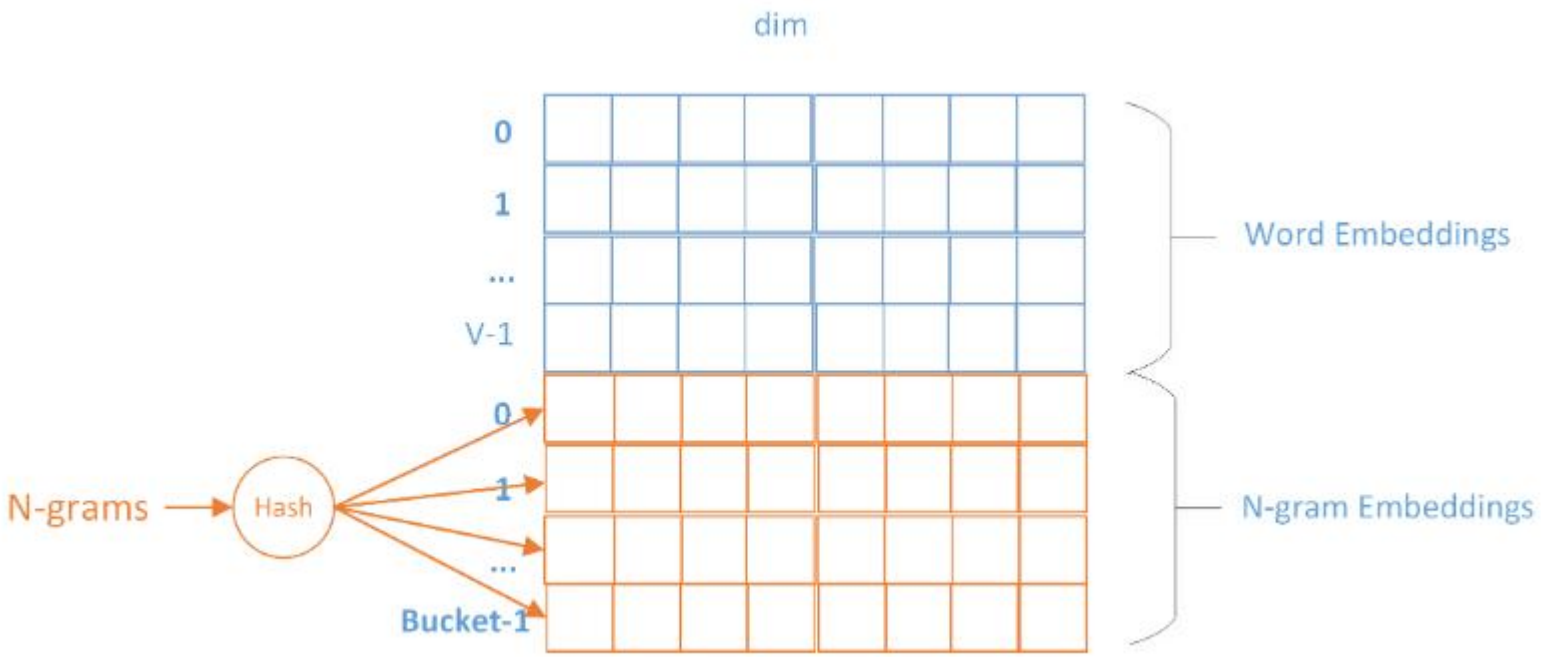
Negative Sampling

NLP基础_词向量_FastText

- FastText在分类中也增加了N-gram的特征，主要是为了通过增加N-Gram的特征信息来保留词序信息(因为隐层是通过简单的求和平均得到的)，比如：某篇文档有3个词，w1、w2、w3，N-gram取N为2，w1、w2、w3以及bigram w12、w23是新的embedding向量，那么文章的隐层表示为：

$$h = \frac{1}{5}(w_1 + w_2 + w_3 + w_{12} + w_{23})$$

NLP基础_词向量_FastText



NLP基础_词向量_cw2vec

- cw2vec(Learning Chinese Word Embeddings with Stroke n-gram Information)
  - CCKS2018阿里健康团队中文电子病历命名实体识别评测任务冠军
  - 思想：**类似FastText的思想，利用中文汉字的笔画信息，使用N-Gram的的方式来提取中文汉字对应的高阶特征信息。**
  - http://www.statnlp.org/wp-content/uploads/papers/2018/cw2vec/cw2vec.pdf

NLP基础_词向量_cw2vec

字符为单位

部首为单位

NLP基础_词向量_cw2vec

- 可以看到从偏旁部首或者字件来提取词语的信息可以改进基于汉字的词语信息的提取效果，但是在某些汉字中，偏旁的设计仅只是为了方便汉字的查询，而不是为了表达汉字的语义信息，所以在cw2vec中提出了一种基于笔画的特征信息提取。

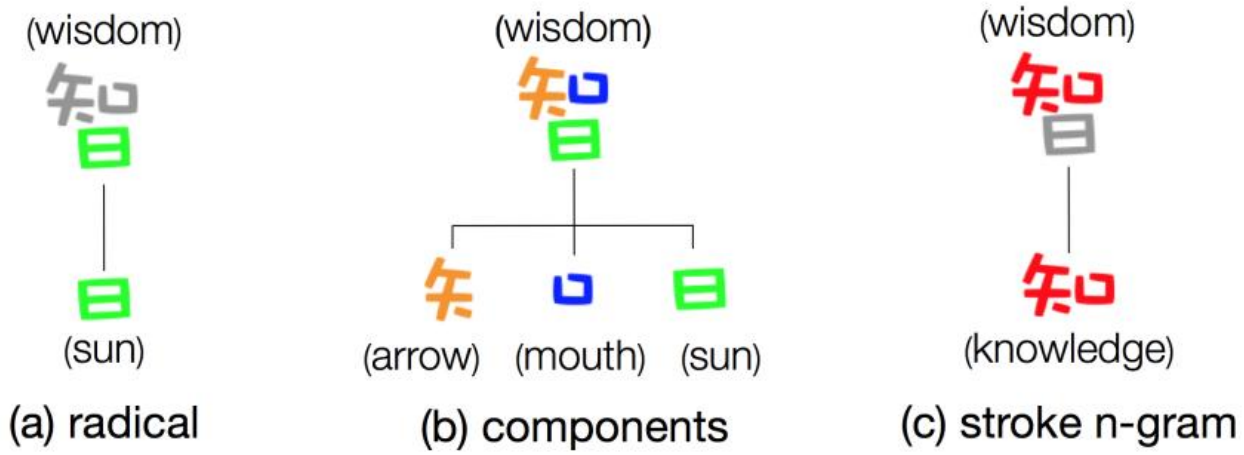


Figure 1: Radical v.s. components v.s. stroke n-gram

NLP基础_词向量_cw2vec

- 1. 将词语分割成字符；
- 2. 提取每个字符的笔画信息(查映射表)，然后将所有字符的笔画信息组合到一起；
- 3. 查表得到每个笔画对应的ID，组成这个词语对应的ID列表；
- 4. 产生N-Gram笔画特征。

Stroke Name	Horizontal	Vertical	Left-falling	Right-falling	Turning
Shape, ID	一 (㇀), 1	丨(亅), 2	丿, 3	乀 (丶), 4	乛(㇆), 5

Figure 3: General shapes of Chinese strokes.

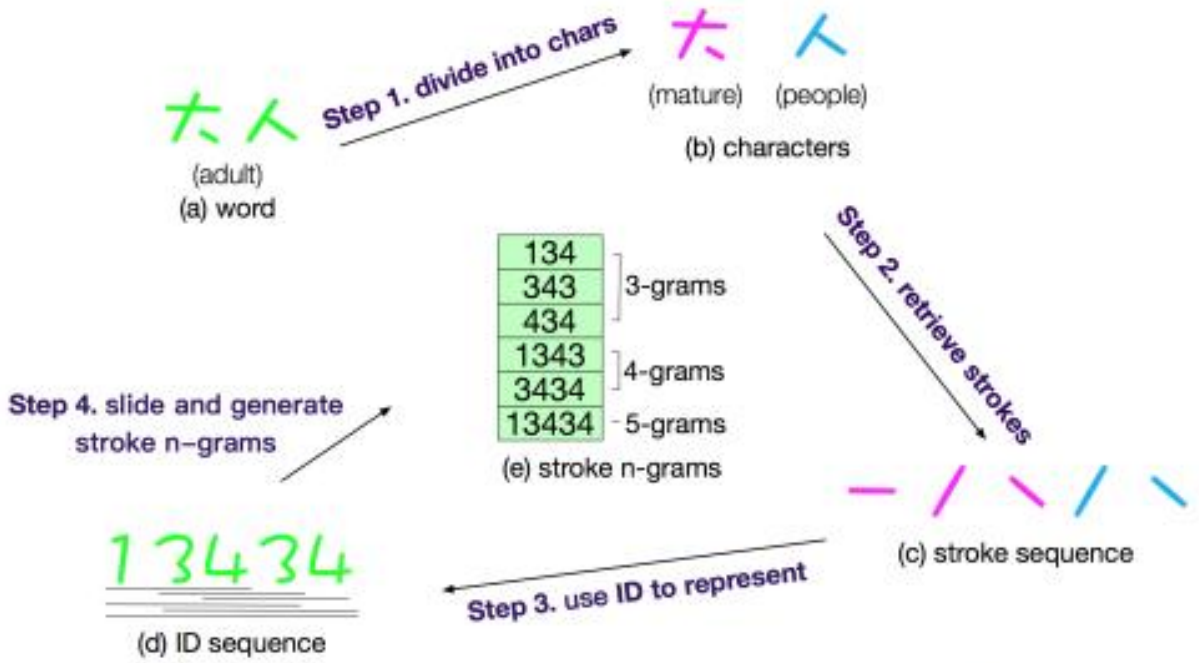


Figure 4: An illustrative example to show the procedures of the generation of stroke n-grams from a word.

NLP基础_词向量_cw2vec

- cw2vec使用和Word2Vec中的Skip-Gram的基础上进行模型训练，仅仅是将词语替换为词语的n-gram笔画特征信息来进行模型训练。
  - 短语：治理 雾霾 刻不容缓
  - 中心词：雾霾
  - 上下文词：治理 刻不容缓

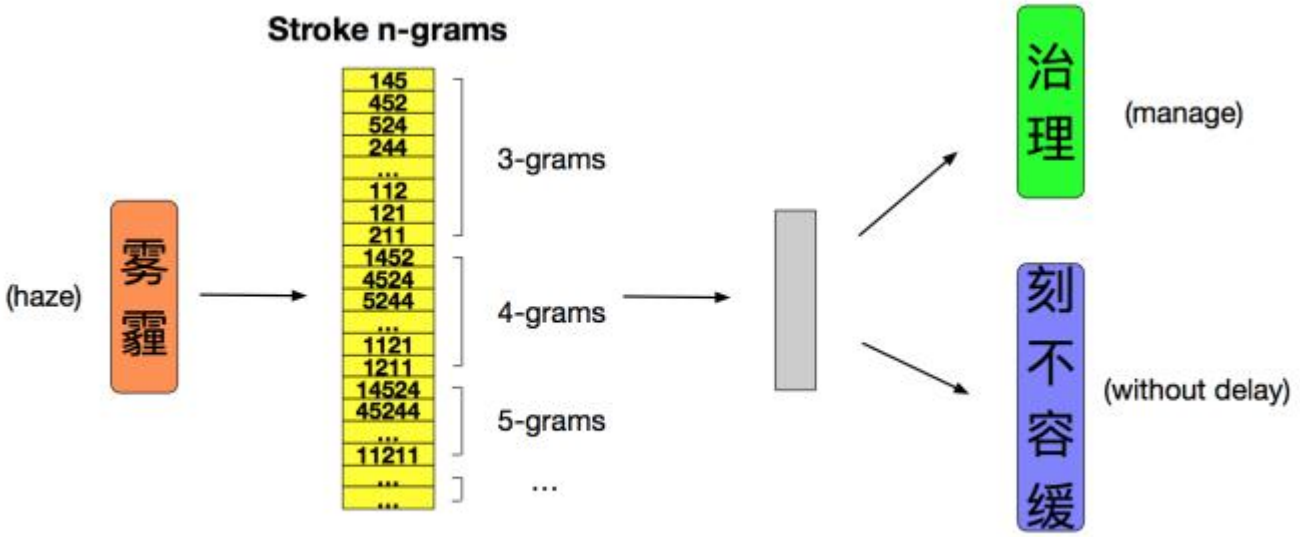


Figure 2: The overall architecture of our approach.

NLP基础_词向量_cw2vec

- 模型损失函数如下：

$$sim(w,c) = \sum_{q \in S(w)} \vec{q} \cdot \vec{c}$$

$$\mathcal{L} = \sum_{w \in D} \sum_{c \in T(w)} \log \sigma(sim(w,c)) + \lambda \mathbb{E}_{c' \sim P}[\log \sigma(-sim(w,c'))]$$

NLP基础_词向量_cw2vec

- 模型损失函数如下：

$$\mathcal{L} = \sum_{w \in D} \sum_{c \in T(w)} log\sigma(sim(w,c)) + \sum_{i=1}^{\lambda} \mathbb{E}_{c' \sim P(D)} [\![log\sigma(-sim(w,c'))]\!]$$

其中，W和C分别为当前词语和上下文词语，σ是sigmoid函数，T(w)是当前词语划窗内的所有词语集合，D是训练语料的全部文本。为了避免传统softmax带来的巨大计算量，这篇论文也采用了负采样的方式。C'为随机选取的词语，称为“负样例”，λ是负样例的个数，而$\mathbb{E}_{c' \sim P(D)} [\![\cdot]\!]$则表示负样例C'按照词频分布进行的采样，其中语料中出现次数越多的词语越容易被采样到。相似性sim(·,·)函数被按照如下构造：

$$sim(w,c) = \sum_{q \in S(w)} \vec{q} \cdot \vec{c}$$

其中，$\vec{q}$为当前词语对应的一个n元笔画向量，而$\vec{c}$是其对应的上下文词语的词向量。这项技术将当前词语拆解为其对应的n元笔画，但保留每一个上下文词语不进行拆解。S(w)为词语w所对应的n元笔画的集合。在算法执行前，这项研究先扫描每一个词语，生成n元笔画集合，针对每一个n元笔画，都有对应的一个n元笔画向量，在算法开始之前做随机初始化，其向量维度和词向量的维度相同。

NLP基础_词向量_cw2vec

Model	Word Similarity		Word Analogy		Text Classification	Named Entity Recognition
	wordsim-240	wordsim-296	3CosAdd	3CosMul		
skip-gram (Mikolov et al. 2013b)	44.2	44.4	58.3	58.9	93.4	65.1
cbow (Mikolov et al. 2013b)	47.0	50.2	54.3	53.5	93.4	59.6
GloVe (Pennington, Socher, and Manning 2014)	45.2	44.3	68.8	66.7	94.2	66.0
CWE (Chen et al. 2015)	50.0	51.5	68.5	69.6	93.2	65.8
GWE (Su and Lee 2017)	50.0	49.1	50.8	50.6	94.3	65.5
JWE (Xin and Song 2017)	48.0	**52.7**	74.2	76.3	94.2	67.9
`cw2vec` (stroke n-grams)	**50.4**	**52.7**	**78.1**	**80.5**	**95.3**	**71.7**

Table 1: Performance on word similarity, word analogy task, text classification and named entity recognition. The embeddings are set as 300 dimensions. The evaluation metric is $\rho \times 100$ for word similarity, accuracy percentage for word analogy and text classification, $F1$-measure for named entity recognition task.

wordsim-240/wordsim-296：词相似度数据集
3CosAdd/3CosMul：词对比任务数据集
Glove：2014年斯坦福发表的词向量生成方式
CWE：2015清华大学中文汉字词向量生成方式
GWE：2017年台湾大学在CWE的基础上汉字词向量生成方式
JWE：2017年香港科技大学基于CBOW的汉字词向量生成方式

NLP基础_词向量_cw2vec

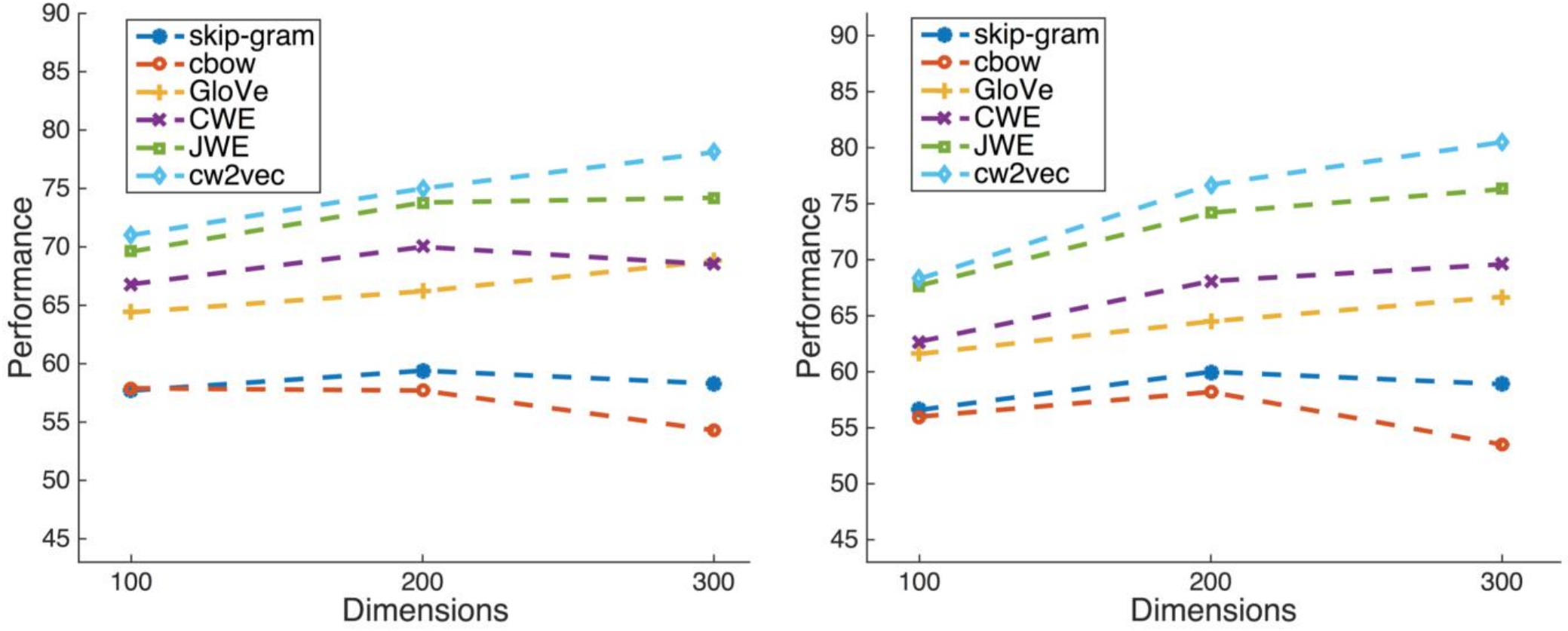


Figure 5: Performance on word analogy over dimensions

NLP基础_词向量_cw2vec

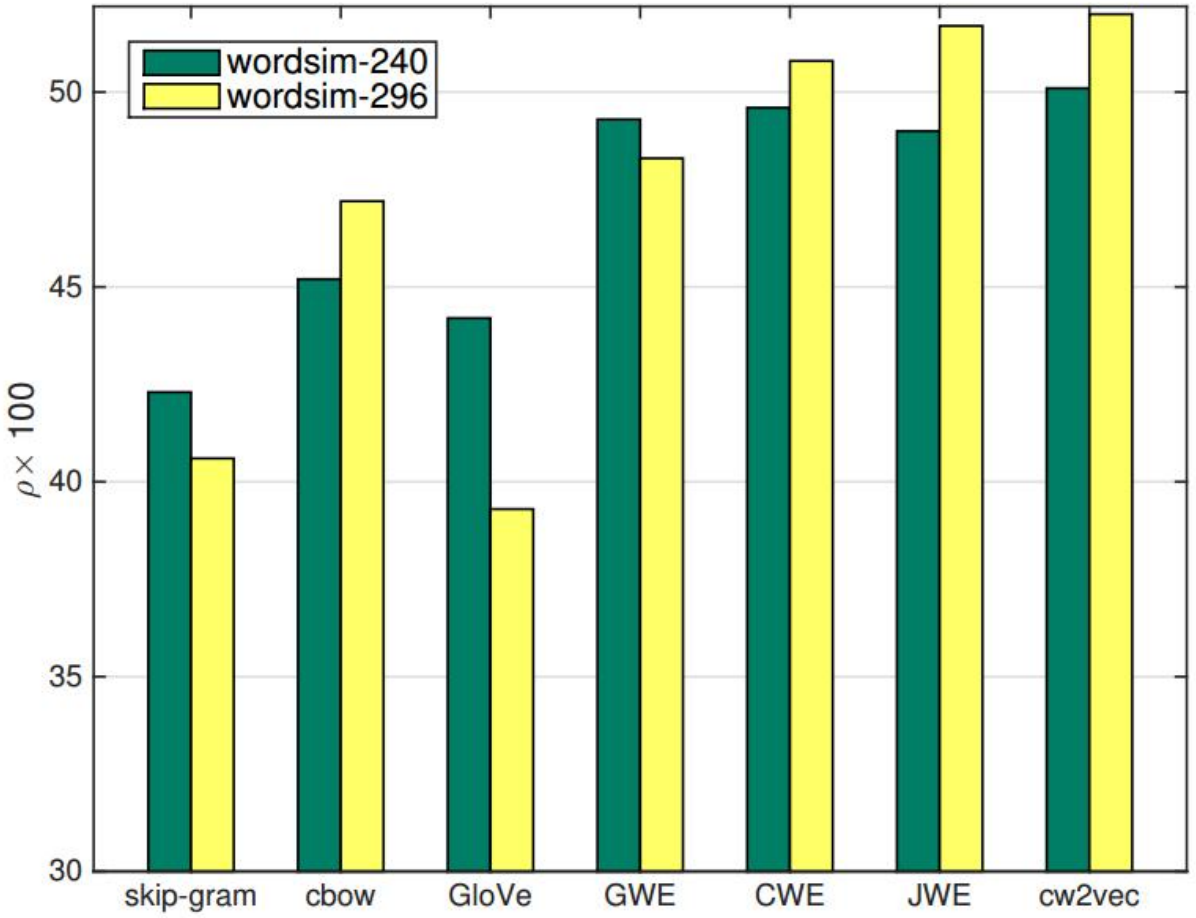


Figure 6: Performance on word similarity, trained on the front 20% wikipedia articles. The embeddings are set as 100 dimensions.

cw2vec实现参考：https://github.com/zhang2010hao/cw2vec-pytorch

THANKS!