火光摇曳

夜幕降临之际，火光摇曳妩媚、灿烂多姿，是最美最美的… …

语义分析的一些方法(一)

2015/02/04 机器学习、自然语言处理、计算广告学 vincentyao

语义分析，本文指运用各种机器学习方法，挖掘与学习文本、图片等的深层次概念。wikipedia上的解释：In machine learning, semantic analysis of a corpus is the task of building structures that approximate concepts from a large set of documents(or images)。

工作这几年，陆陆续续实践过一些项目，有搜索广告，社交广告，微博广告，品牌广告，内容广告等。要使我们广告平台效益最大化，首先需要理解用户，Context(将展示广告的上下文)和广告，才能将最合适的广告展示给用户。而这其中，就离不开对用户，对上下文，对广告的语义分析，由此催生了一些子项目，例如文本语义分析，图片语义理解，语义索引，短串语义关联，用户广告语义匹配等。

接下来我将写一写我所认识的语义分析的一些方法，虽说我们在做的时候，效果导向居多，方法理论理解也许并不深入，不过权当个人知识点总结，有任何不当之处请指正，谢谢。

本文主要由以下四部分组成：文本基本处理，文本语义分析，图片语义分析，语义分析小结。先讲述文本处理的基本方法，这构成了语义分析的基础。接着分文本和图片两节讲述各自语义分析的一些方法，值得注意的是，虽说分为两节，但文本和图片在语义分析方法上有很多共通与关联。最后我们简单介绍下语义分析在广点通“用户广告匹配”上的应用，并展望一下未来的语义分析方法。

1 文本基本处理

在讲文本语义分析之前，我们先说下文本基本处理，因为它构成了语义分析的基础。而文本处理有很多方面，考虑到本文主题，这里只介绍中文分词以及Term Weighting。

1.1 中文分词

拿到一段文本后，通常情况下，首先要做分词。分词的方法一般有如下几种：

- 基于字符串匹配的分词方法。此方法按照不同的扫描方式，逐个查找词库进行分词。根据扫描方式可细分为：正向最大匹配，反向最大匹配，双向最大匹配，最小切分(即最短路径)；总之就是各种不同的启发规则。
- 全切分方法。它首先切分出与词库匹配的所有可能的词，再运用统计语言模型决定最优的切分结果。它的优点在于可以解决分词中的歧义问题。下图是一个示例，对于文本串“南京市长江大桥”，首先进行词条检索(一般用Trie存储)，找到匹配的所有词条（南京，市，长江，大桥，南京市，长江大桥，市长，江大桥，江大，桥），以词网格(word lattices)形式表示，接着做路径搜索，基于统计语言模型(例如n-gram)[18]找到最优路径，最后可能还需要命名实体识别。下图中

“南京市 长江 大桥”的语言模型得分，即P(南京市，长江，大桥)最高，则为最优切分。

```
南京 市 长江 大桥 </s>  : -11.468807
南京市 长江大桥 </s>  : -10.098166
南京市 长江 大桥 </s>  : -8.224921
南京 市长 江大桥 </s>  : -12.734082
南京 市长 江大 桥 </s>  : -15.659960
```

图1. “南京市长江大桥”语言模型得分

- 由字构词的分词方法。可以理解为字的分类问题，也就是自然语言处理中的sequence labeling问题，通常做法里利用HMM，MAXENT，MEMM，CRF等预测文本串每个字的tag[62]，譬如B，E，I，S，这四个tag分别表示：beginning, inside, ending, single，也就是一个词的开始，中间，结束，以及单个字的词。 例如“南京市长江大桥”的标注结果可能为：“南(B)京(I)市(E)长(B)江(E)大(B)桥(E)”。由于CRF既可以像最大熵模型一样加各种领域feature，又避免了HMM的齐次马尔科夫假设，所以基于CRF的分词目前是效果最好的，具体请参考文献[61,62,63]。除了HMM，CRF等模型，分词也可以基于深度学习方法来做，如文献[9][10]所介绍，也取得了state-of-the-art的结果。

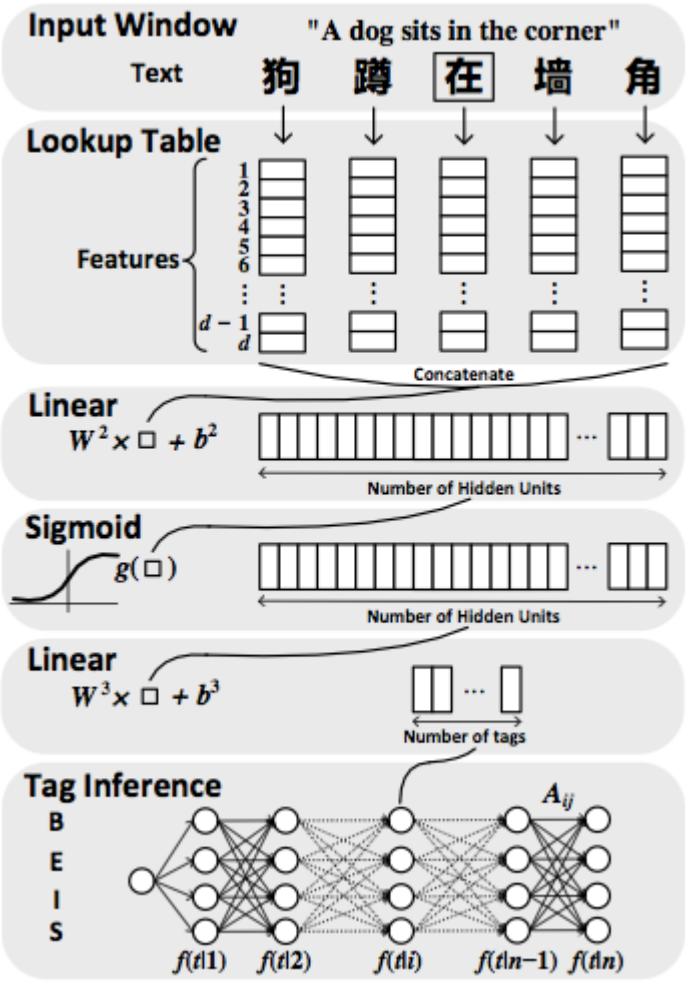


图2. 基于深度学习的中文分词

上图是一个基于深度学习的分词示例图。我们从上往下看，首先对每一个字进行Lookup Table，映射到一个固定长度的特征向量(这里可以利用词向量，boundary entropy，accessor variety等)；接着经过一个标准的神经网络，分别是linear，sigmoid，linear层，对于每个字，预测该字属于B,E,I,S的概率；最后输出是一个矩阵，矩阵的行是B,E,I,S 4个tag，利用viterbi算法就可以完

成标注推断，从而得到分词结果。

一个文本串除了分词，还需要做词性标注，命名实体识别，新词发现等。通常有两种方案，一种是pipeline approaches，就是先分词，再做词性标注；另一种是joint approaches，就是把这些任务用一个模型来完成。有兴趣可以参考文献[9][62]等。

一般而言，方法一和方法二在工业界用得比较多，方法三因为采用复杂的模型，虽准确率相对高，但耗时较大。

1.2 语言模型

前面在讲“全切分分词”方法时，提到了语言模型，并且通过语言模型，还可以引出词向量，所以这里把语言模型简单阐述一下。

语言模型是用来计算一个句子产生概率的概率模型，即P(w_1,w_2,w_3…w_m)，m表示词的总个数。根据贝叶斯公式：P(w_1,w_2,w_3 … w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1,w_2) … P(w_m|w_1,w_2 … w_{m-1})。

最简单的语言模型是N-Gram，它利用马尔科夫假设，认为句子中每个单词只与其前n–1个单词有关，即假设产生w_m这个词的条件概率只依赖于前n–1个词，则有P(w_m|w_1,w_2…w_{m-1}) = P(w_m|w_{m-n+1},w_{m-n+2} … w_{m-1})。其中n越大，模型可区别性越强，n越小，模型可靠性越高。

N-Gram语言模型简单有效，但是它只考虑了词的位置关系，没有考虑词之间的相似度，词语法和词语义，并且还存在数据稀疏的问题，所以后来，又逐渐提出更多的语言模型，例如Class-based ngram model，topic-based ngram model，cache-based ngram model，skipping ngram model，指数语言模型（最大熵模型，条件随机域模型）等。若想了解更多请参考文章[18]。

最近，随着深度学习的兴起，神经网络语言模型也变得火热[4]。用神经网络训练语言模型的经典之作，要数Bengio等人发表的《A Neural Probabilistic Language Model》[3]，它也是基于N-Gram的，首先将每个单词w_{m-n+1},w_{m-n+2} … w_{m-1}映射到词向量空间，再把各个单词的词向量组合成一个更大的向量作为神经网络输入，输出是P(w_m)。本文将此模型简称为ffnnlm（Feed-forward Neural Net Language Model）。ffnnlm解决了传统n-gram的两个缺陷：(1)词语之间的相似性可以通过词向量来体现；(2)自带平滑功能。文献[3]不仅提出神经网络语言模型，还顺带引出了词向量，关于词向量，后文将再细述。

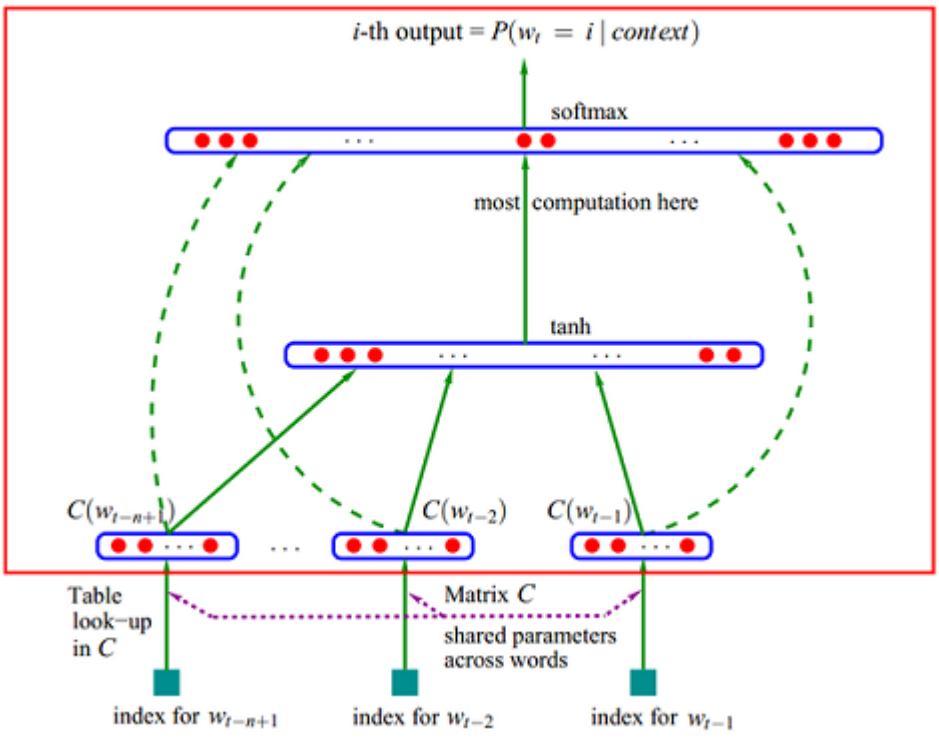


图3. 基于神经网络的语言模型

从最新文献看，目前state-of-the-art语言模型应该是基于循环神经网络(recurrent neural network)的语言模型，简称rnnlm[5][6]。循环神经网络相比于传统前馈神经网络，其特点是：可以存在有向环，将上一次的输出作为本次的输入。而rnnlm和ffnnlm的最大区别是：ffnnmm要求输入的上下文是固定长度的，也就是说n-gram中的 n 要求是个固定值，而rnnlm不限制上下文的长度，可以真正充分地利用所有上文信息来预测下一个词，本次预测的中间隐层信息(例如下图中的context信息)可以在下一次预测里循环使用。

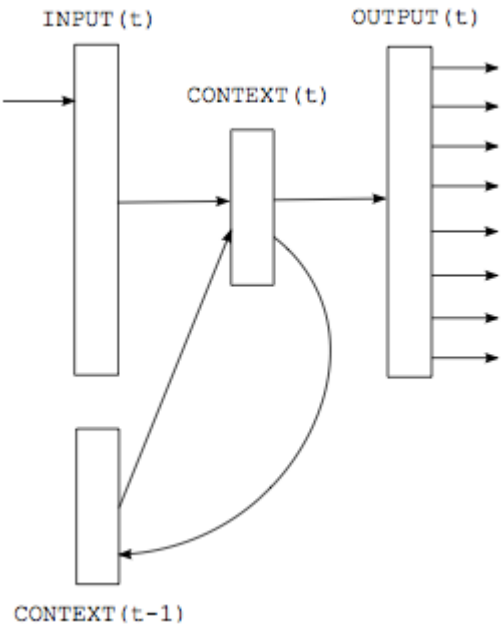


图4. 基于simple RNN(time-delay neural network)的语言模型

如上图所示，这是一个最简单的rnnlm，神经网络分为三层，第一层是输入层，第二层是隐藏层(也叫context层)，第三层输出层。 假设当前是t时刻，则分三步来预测P(w_m)：

- 单词w_{m-1}映射到词向量，记作input(t)
- 连接上一次训练的隐藏层context(t–1)，经过sigmoid function，生成当前t时刻的context(t)
- 利用softmax function，预测P(w_m)

参考文献[7]中列出了一个rnnlm的library，其代码紧凑。利用它训练中文语言模型将很简单，上面“南京市 长江 大桥”就是rnnlm的预测结果。

基于RNN的language model利用BPTT(BackPropagation through time)算法比较难于训练，原因就是深度神经网络里比较普遍的vanishing gradient问题[55]（在RNN里，梯度计算随时间成指数倍增长或衰减，称之为Exponential Error Decay）。所以后来又提出基于LSTM(Long short term memory)的language model，LSTM也是一种RNN网络，关于LSTM的详细介绍请参考文献[54,49,52]。LSTM通过网络结构的修改，从而避免vanishing gradient问题。

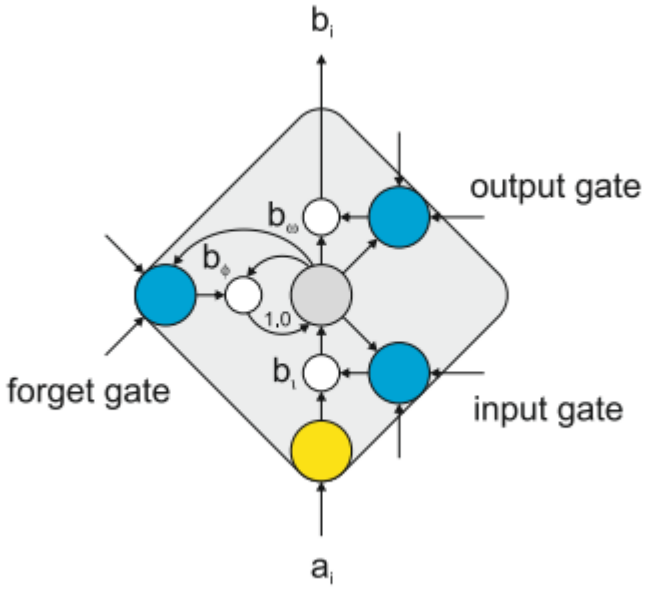


图5. LSTM memory cell

如上图所示，是一个LSTM unit。如果是传统的神经网络unit，output activation bi = activation_function(ai)，但LSTM unit的计算相对就复杂些了，它保存了该神经元上一次计算的结果，通过input gate，output gate，forget gate来计算输出，具体过程请参考文献[53，54]。

1.3 Term Weighting

Term重要性

对文本分词后，接下来需要对分词后的每个term计算一个权重，重要的term应该给与更高的权重。举例来说，“什么产品对减肥帮助最大？”的term weighting结果可能是: “什么 0.1，产品 0.5，对 0.1，减肥

0.8，帮助 0.3，最大 0.2”。Term weighting在文本检索，文本相关性，核心词提取等任务中都有重要作用。

- Term weighting的打分公式一般由三部分组成：local，global和normalization [1,2]。即TermWeight=L_{i,j} G_i N_j。L_{i,j}是term i在document j中的local weight，G_i是term i的global weight，N_j是document j的归一化因子。
  常见的local，global，normalization weight公式[2]有：

Formula	Name	Abbr.
$\begin{matrix}1 & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0\end{matrix}$	Binary	BNRY
f_{ij}	Within-document frequency	FREQ
$\begin{matrix}1 + \log f_{ij} & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0\end{matrix}$	Log	LOGA
$\begin{matrix}\frac{1+\log f_{ij}}{1+\log a_j} & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0\end{matrix}$	Normalized log	LOGN
$\begin{matrix}0.5 + 0.5\left(\frac{f_{ij}}{x_j}\right) & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0\end{matrix}$	Augmented normalized term frequency	ATF1

图6. Local weight formulas

Formula	Name	Abbr.
$\log\left(\frac{N}{n_i}\right)$	Inverse document frequency	IDFB
$\log\left(\frac{N-n_i}{n_i}\right)$	Probabilistic inverse	IDFP
$1 + \sum_{j=1}^{N} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log N}$	Entropy	ENPY
$\frac{F_i}{n_i}$	Global frequency IDF	IGFF
1	No global weight	NONE

图7. Global weight formulas

Formula	Name	Abbr.
$\frac{1}{\sqrt{\sum_{i=0}^{m}(G_i L_{ij})^2}}$	Cosine normalization	COSN
$\frac{1}{(1-slope)+slope\ l_j}$	Pivoted unique normalization	PUQN
1	None	NONE

图8. Normalization factors

Tf-Idf是一种最常见的term weighting方法。在上面的公式体系里，Tf-Idf的local weight是FREQ，glocal weight是IDFB，normalization是None。tf是词频，表示这个词出现的次数。df是文档频率，表示这个词在多少个文档中出现。idf则是逆文档频率，idf=log(TD/df)，TD表示总文档数。Tf-Idf在很多场合都很有效，但缺点也比较明显，以“词频”度量重要性，不够全面，譬如在搜索广告的关键词匹配时就不够用。

除了TF-IDF外，还有很多其他term weighting方法，例如Okapi，MI，LTU，ATC，TF-ICF[59]等。通过local，global，normalization各种公式的组合，可以生成不同的term weighting计算方法。不过上面这些方法都是无监督计算方法，有一定程度的通用性，但在一些特定场景里显得不够灵活，不够准确，所以可以基于有监督机器学习方法来拟合term weighting结果。

Okapi	$w_{ij}=\left(\frac{f_{ij}}{0.5+1.5\times\frac{dl}{avg\_dl}+f_{ij}}\right)\log\left(\frac{N-n_j+0.5}{f_{ij}+0.5}\right)$

图9. Okapi计算公式

- 利用有监督机器学习方法来预测weight。这里类似于机器学习的分类任务，对于文本串的每个term，预测一个[0,1]的得分，得分越大则term重要性越高。既然是有监督学习，那么就需要训练数据。如果采用人工标注的话，极大耗费人力，所以可以采用训练数据自提取的方法，利用程序从搜索日志里自动挖掘。从海量日志数据里提取隐含的用户对于term重要性的标注，得到的训练数据将综合亿级用户的“标注结果”，覆盖面更广，且来自于真实搜索数据，训练结果与标注的目标集分布接近，训练数据更精确。下面列举三种方法(除此外，还有更多可以利用的方法)：
  - 从搜索session数据里提取训练数据，用户在一个检索会话中的检索核心意图是不变的，提取出核心意图所对应的term，其重要性就高。
  - 从历史短串关系资源库里提取训练数据，短串扩展关系中，一个term出现的次数越多，则越重要。
  - 从搜索广告点击日志里提取训练数据，query与bidword共有term的点击率越高，它在query中的重要程度就越高。

  通过上面的方法，可以提取到大量质量不错的训练数据（数十亿级别的数据，这其中可能有部分样本不准确，但在如此大规模数据情况下，绝大部分样本都是准确的）。

有了训练数据，接下来提取特征，基于逻辑回归模型来预测文本串中每个term的重要性。所提取的特征包括：

- term的自解释特征，例如term专名类型，term词性，term idf，位置特征，term的长度等；
- term与文本串的交叉特征，例如term与文本串中其他term的字面交叉特征，term转移到文本串中其他term的转移概率特征，term的文本分类、topic与文本串的文本分类、topic的交叉特征等。

核心词、关键词提取

- 短文本串的核心词提取。对短文本串分词后，利用上面介绍的term weighting方法，获取term weight后，取一定的阈值，就可以提取出短文本串的核心词。
- 长文本串(譬如web page)的关键词提取。这里简单介绍几种方法。想了解更多，请参考文献[69]。
  - 采用基于规则的方法。考虑到位置特征，网页特征等。
  - 基于广告主购买的bidword和高频query建立多模式匹配树，在长文本串中进行全字匹配找出候选关键词，再结合关键词weight，以及某些规则找出优质的关键词。
  - 类似于有监督的term weighting方法，也可以训练关键词weighting的模型。
  - 基于文档主题结构的关键词抽取，具体可以参考文献[71]。

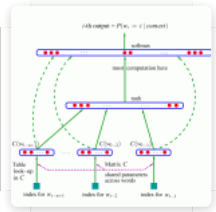
参考文献

1. Term-weighting approaches in automatic text retrieval，Gerard Salton et.
2. New term weighting formulas for the vector space method in information retrieval
3. A neural probabilistic language model 2003
4. Deep Learning in NLP-词向量和语言模型
5. Recurrent neural network based language models
6. Statistical Language Models based on Neural Networks，mikolov博士论文
7. Rnnlm library
8. A survey of named entity recognition and classification
9. Deep learning for Chinese word segmentation and POS tagging
10. Max-margin tensor neural network for chinese word segmentation
11. Learning distributed representations of concepts
12. Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements
13. LightLda
14. word2vec
15. Efficient Estimation of Word Representations in Vector Space
16. Deep Learning实战之word2vec
17. word2vec中的数学原理详解 出处2
18. 斯坦福课程-语言模型
19. Translating Videos to Natural Language Using Deep Recurrent Neural Networks
20. Distributed Representations of Sentences and Documents
21. Convolutional Neural Networks卷积神经网络
22. A New, Deep-Learning Take on Image Recognition

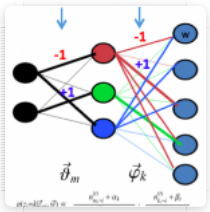
23. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
24. A Deep Learning Tutorial: From Perceptrons to Deep Networks
25. Deep Learning for Computer Vision
26. Zero-shot leanring by convex combination of semantic embeddings
27. Sequence to sequence learning with neural network
28. Exploting similarities among language for machine translation
29. Grammar as Foreign Language Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton, arXiv 2014
30. Deep Semantic Embedding
31. 张家俊. DNN Applications in NLP
32. Deep learning for natural language processing and machine translation
33. Distributed Representations for Semantic Matching
34. distributed_representation_nlp
35. Deep Visual-Semantic Alignments for Generating Image Descriptions
36. Convolutional Neural Networks for Sentence Classification
37. Senna
38. ImageNet Large Scale Visual Recognition Challenge
39. Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks
40. Gradient-Based Learning Applied to Document Recognition
41. Effetive use of word order for text categorization with convolutional neural network， Rie Johnson
42. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation
43. Show and Tell: A Neural Image Caption Generator
44. Deep Image: Scaling up Image Recognition
45. Large-Scale High-Precision Topic Modeling on Twitter
46. A. Krizhevsky. One weird trick for parallelizing convolutional neural networks. arXiv:1404.5997, 2014
47. A Brief Overview of Deep Learning
48. Going deeper with convolutions. Christian Szegedy. Google Inc. 阅读笔记
49. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling
50. Semi-Supervised Learning Tutorial
51. http://www.zhihu.com/question/24904450
52. LONG SHORT-TERM MEMORY BASED RECURRENT NEURAL NETWORK ARCHITECTURES FOR LARGE VOCABULARY SPEECH RECOGNITION
53. LSTM Neural Networks for Language Modeling
54. LONG SHORT-TERM MEMORY
55. Bengio, Y., Simard, P., Frasconi, P., “Learning long-term dependencies with gradient descent is difficult” IEEE Transactions on Neural Networks 5 (1994), pp. 157–166
56. AliasLDA
57. Gibbs sampling for the uninitiated
58. Learning classifiers from only positive and unlabeled data
59. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams
60. LDA数学八卦
61. Chinese Word Segmentation and Named Entity Recognition Based on Conditional Random

Fields Models
62. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data
63. Chinese Segmentation and New Word Detection using Conditional Random Fields
64. Gregor Heinrich. Parameter estimation for text analysis
65. Peacock：大规模主题模型及其在腾讯业务中的应用
66. L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In KDD, 2009.
67. David Newman. Distributed Algorithms for Topic Models
68. Xuemin. LDA工程实践之算法篇
69. Brian Lott. Survey of Keyword Extraction Techniques
70. Yi Wang, Xuemin Zhao, Zhenlong Sun, Hao Yan, Lifeng Wang, Zhihui Jin, Liubin Wang, Yang Gao, Ching Law, and Jia Zeng. Peacock: Learning Long-Tail Topic Features for Industrial Applications. TIST'2015.
71. 刘知远. 基于文档主题结构的关键词抽取方法研究
72. Hinton. Reducing the Dimensionality of Data with Neural Networks
73. Samaneh Moghaddam. On the design of LDA models for aspect-based opinion mining;
74. The FLDA model for aspect-based opinion mining: addressing the cold start problem
75. Ross Girshick et. Rich feature hierarchies for accurate object detection and semantic segmentation
76. J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013.
77. Baidu/UCLA: Explain Images with Multimodal Recurrent Neural Networks
78. Toronto: Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models
79. Berkeley: Long-term Recurrent Convolutional Networks for Visual Recognition and Description
80. Xinlei Chen et. Learning a Recurrent Visual Representation for Image Caption Generation
81. Hao Fang et. From Captions to Visual Concepts and Back
82. Modeling Documents with a Deep Boltzmann Machine
83. A Deep Dive into Recurrent Neural Nets
84. Xiang zhang et. Text Understanding from Scratch

相关文章

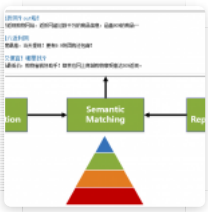


2015/03/15
[我们是这样理解语言的-3]神经网络语言模型

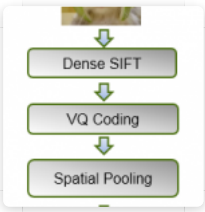


2015/02/04
语义分析的一些方法(二)

2015/01/28
Peacock: 大规模主题模型及其在腾讯业务中的应用



2014/06/19
[我们是这样理解语言的-1]文本分析平台TextMiner



2015/02/04
语义分析的一些方法(三)

分享到...

1 条评论 最新 最早 最热

杯面

学习了，楼主总结的不错

2015年7月17日 回复

顶 转发

社交帐号登录:

微信 微博 QQ 人人 更多»

说点什么吧...

发布

火光摇曳正在使用多说