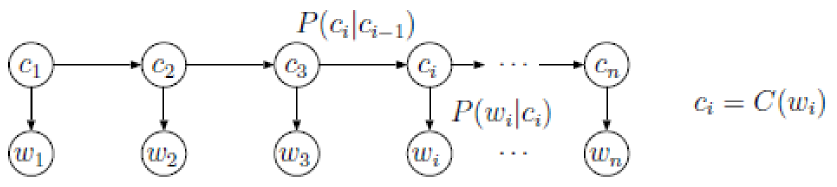
1、 Single word vectors:

(1) Distributional Techniques

(2) Brown Clusters(布朗聚类)

http://blog.csdn.net/u014516670/article/details/50574147

布朗聚类是一种自底向上的层次聚类算法，基于 n-gram 模型和马尔科夫链模型。布朗聚类是一种硬聚类，每一个词都在且只在唯一的一个类中。



w 是词，c 是词所属的类。

布朗聚类的输入是一个语料库，这个语料库是一个词序列，输出是一个二叉树，树的叶子节点是一个个词，树的中间节点是我们想要的类（中间结点作为根节点的子树上的所有叶子为类中的词）。

(3) Useful as features as inside models

(4) Cannot capture longer phrases

2、 Document Vectors

(1) Bags of words models

(2) PCA(LSA, LDA)

(3) Great for IR, document exploration, etc.

(4) Ignore word order, no detailed understanding.

我们希望能够找到一种方法把短语、句子的语义用向量表示出来，同时保存词语顺序信息。

3、 How map phrases into a vector space?

Use principle of compositionality(用组合的原理)：

The meaning of a sentence is determined by the meanings of its words and the rules combine them.

4、 PAPER Parsing with Compositional Vector Grammars

Compositional Vector Grammar:

A model jointly find syntactic structure and capture compositional semantic information. CVGs build on two observations. Firstly, that a lot of the structure and regularity in languages can be captured by well-designed syntactic patterns(符号关系模型).Hence, the CVG builds on top of a standard PCFG parser.

The CVG model merges ideas from both generative models(生成模型) that assume discrete syntactic categories and discriminative models(判别模型) that are trained using continuous vectors.

4.1 Word Vector Representations.

Assume we have are given a sentence as an ordered list of m words. Each word w has an index [w] = i into the columns of the embedding matrix. This index is used to retrieve the word's vector representation aw using a simple multiplication with a binary vector e(二进制向量), which is zero everywhere, except at the ith index.

[redacted] So $a_w = Le_i \in \mathbb{R}^n$. Henceforth, after mapping each word to its vector, we represent a sentence S as an ordered list of (word,vector) pairs: $x = ((w_1, a_{w_1}), \ldots, (w_m, a_{w_m}))$.

4.2 Max-Margin Training Objective for CVGs

First define a structured margin loss for predicting a tree for a given correct tree. The loss increases the more incorrect the proposed parse tree is. The discrepancy between trees is measured by counting the number of nodes with an incorrect span in the proposed tree:

$$\Delta(y_i, \hat{y}) = \sum_{d \in N(\hat{y})} \kappa \mathbf{1}\{d \notin N(y_i)\}. \qquad (1)$$

We set k = 0.1 in all experiments.

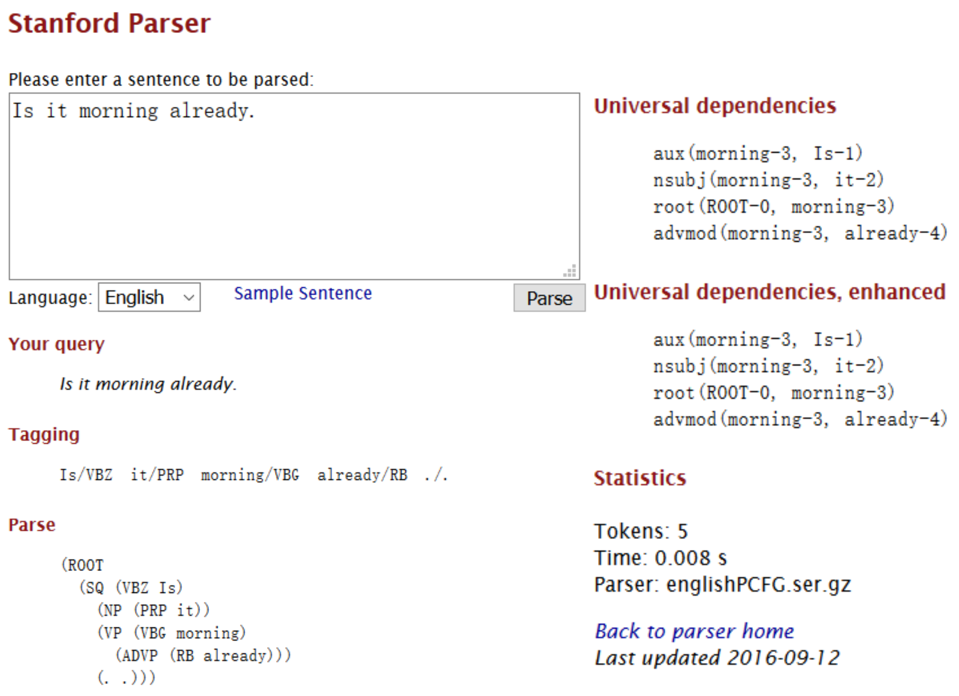
Max-Margin 在第二篇论文详细介绍 Subgradient Methods for Structure Prediction：

Software : Stanford Parser http://nlp.stanford.edu:8080/parser/index.jsp

A natural language parser is a program that works out the grammatical **structure of sentences**, for instance, which groups of words go together (as "phrases") and which words are the **subject** or **object** of a verb.

问题：Stanford Parser 的分析结果看不懂。

例子如下：也可以分析中文。



Stanford Parser

Please enter a sentence to be parsed:

Is it morning already.

Language: English   Sample Sentence   Parse

Your query

Is it morning already.

Tagging

```
Is/VBZ  it/PRP  morning/VBG  already/RB  ./.
```

Parse

```
(ROOT
  (SQ (VBZ Is)
    (NP (PRP it))
    (VP (VBG morning)
      (ADVP (RB already)))
    (. .)))
```

Universal dependencies

```
aux(morning-3, Is-1)
nsubj(morning-3, it-2)
root(ROOT-0, morning-3)
advmod(morning-3, already-4)
```

Universal dependencies, enhanced

```
aux(morning-3, Is-1)
nsubj(morning-3, it-2)
root(ROOT-0, morning-3)
advmod(morning-3, already-4)
```

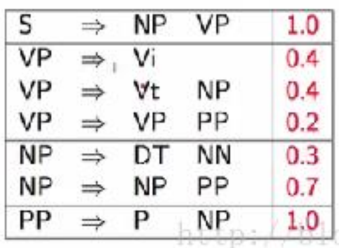
Statistics

Tokens: 5
Time: 0.008 s
Parser: englishPCFG.ser.gz

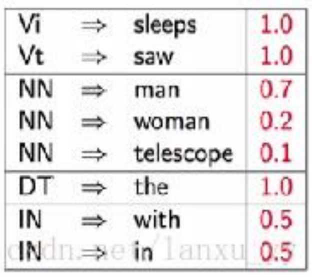
Back to parser home
Last updated 2016-09-12

PCFG: 概率分布的上下文无关语法

一种常见的方法既是 PCFG （Probabilistic Context-Free Grammar）。如下图所示，除了常规的语法规则以外，我们还对每一条规则赋予了一个概率。对于每一棵生成的语法树，我们将其中所以规则的概率的乘积作为语法树的出现概率。当我们获得多颗语法树时，我们可以分别计算每颗语法树的概率 p(t)，出现概率最大的那颗语法树就是我们希望得到的结果，即 argmax p(t)。



S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0



Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5