

# AIDemo

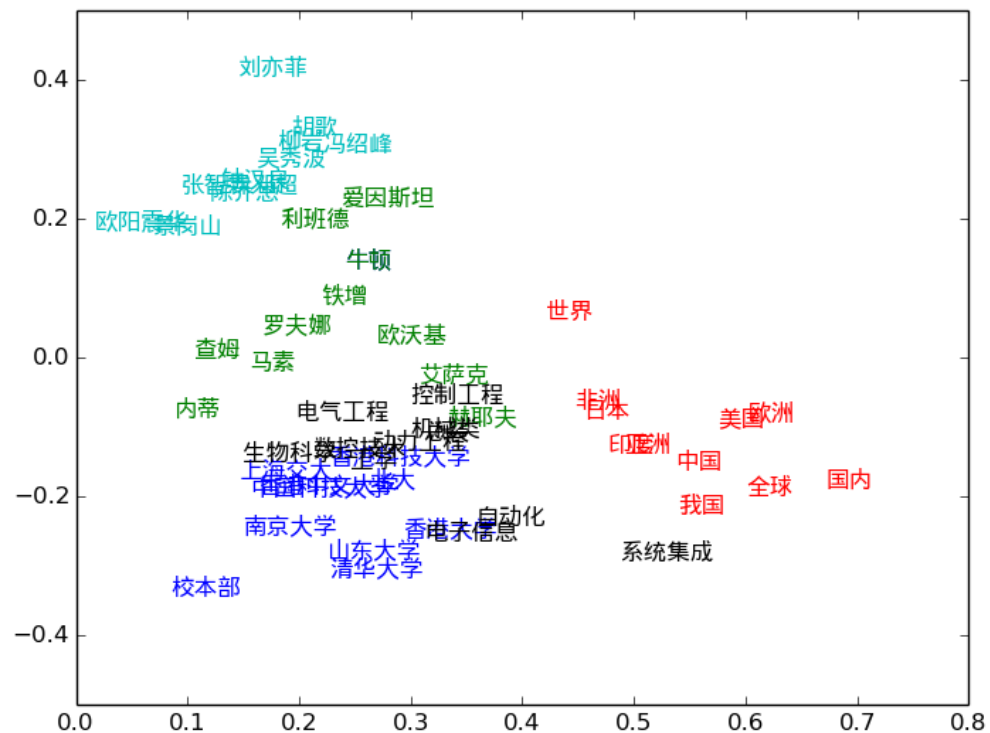


# 理解你的语言

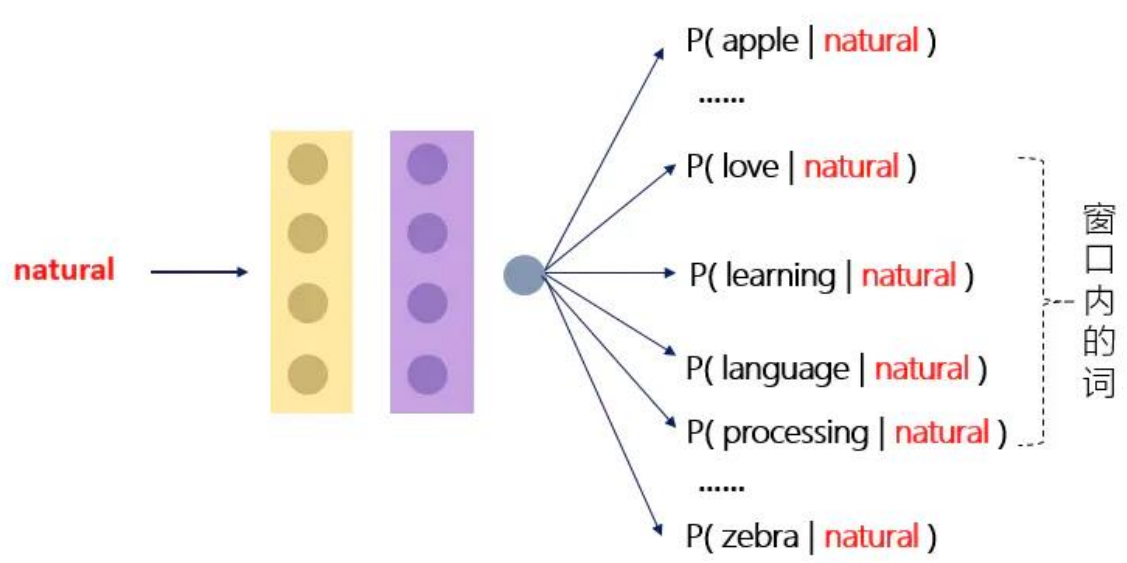
李蓝天

# 词向量

- 词向量是单词在一个连续语义空间上的映射。
- 语义接近的单词所对应的向量间距离更小。
- 语义空间低维、连续、可计算



# 基本思路

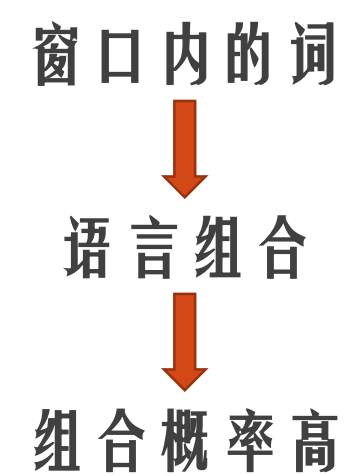


计算窗口内的词的  
概率的乘积：

$$\begin{aligned} &P(\text{love} | \text{natural}) \\ &\times P(\text{learning} | \text{natural}) \\ &\times P(\text{language} | \text{natural}) \\ &\times P(\text{processing} | \text{natural}) \end{aligned}$$

由于窗口内的这几个词是真实的语言组合，  
所以我们希望它们组合的概率尽可能大。

训练准则



# 实验一：生成《射雕英雄传》词向量

- 文本预料：金庸小说《射雕英雄传》
- 文本分词：Jieba 分词工具
- 构建词表：选出高频词汇（出现频率超过 10 次的单词）
- 训练词向量：利用 Word2Vec 工具，生成词向量
- 可视化：利用 t-SNE 工具，将词向量投影到二维空间。

# 实验步骤

- 认真阅读 `word2vce/lang/doc/README`，了解实验步骤。
  - 可在图形界面下双击打开，也可在终端下用 `vim` 打开
  - 关于 `vim` 的操作，请 [aibook.csl.t.org](http://aibook.csl.t.org) 下 Linux Shell 简易教程
- 打开终端，进入 `lang` 的 `code` 目录。
- 运行缺省程序

**sh run.sh**

- 运行上述命令得到基于《射雕英雄传》中所有文本词向量的二维映射图片，关闭图片后自动将图片转为 PDF 保存。

# 运行界面

- 进入到 lang 的子文件夹 code 中运行run.sh



The screenshot shows a terminal window titled "centos7 [正在运行] - Oracle VM VirtualBox". The window has a menu bar with "管理", "控制", "视图", "热键", "设备", and "帮助". Below the menu bar, there are icons for "应用程序", "位置", and "终端". The terminal title bar reads "tutorial@localhost:~/aibook/demo/lang/word2vec/code". The terminal content shows the following commands and output:

```
[tutorial@localhost ~]$ ls
aibook 公共 模板 视频 图片 文档 下载 音乐 桌面
[tutorial@localhost ~]$ cd aibook/demo/lang/word2vec/code/
[tutorial@localhost code]$ sh run.sh
```

Two red callout boxes with white text are present: one labeled "路径" (Path) pointing to the directory path in the terminal, and another labeled "运行" (Run) pointing to the command execution.



# 实验分析

- 从上图可以看出，由于词表数量过大，词向量在图片上显示过于稠密，并不能清晰地显示词之间的关系距离。
- 尝试通过修改 run.sh 中的参数，减少词表数量。如下图所示，将箭头所指的 10 修改为 50，代表是只保留词频大于 50 的词。

```
tutorial@localhost:~/aibook/demo/lang/word2vec/code
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
source /home/tutorial/aibook/demo/env/py2.7/bin/activate
#STEP1: word segmentation
cat txt/shediao.txt | python2 seg.py > seg.txt

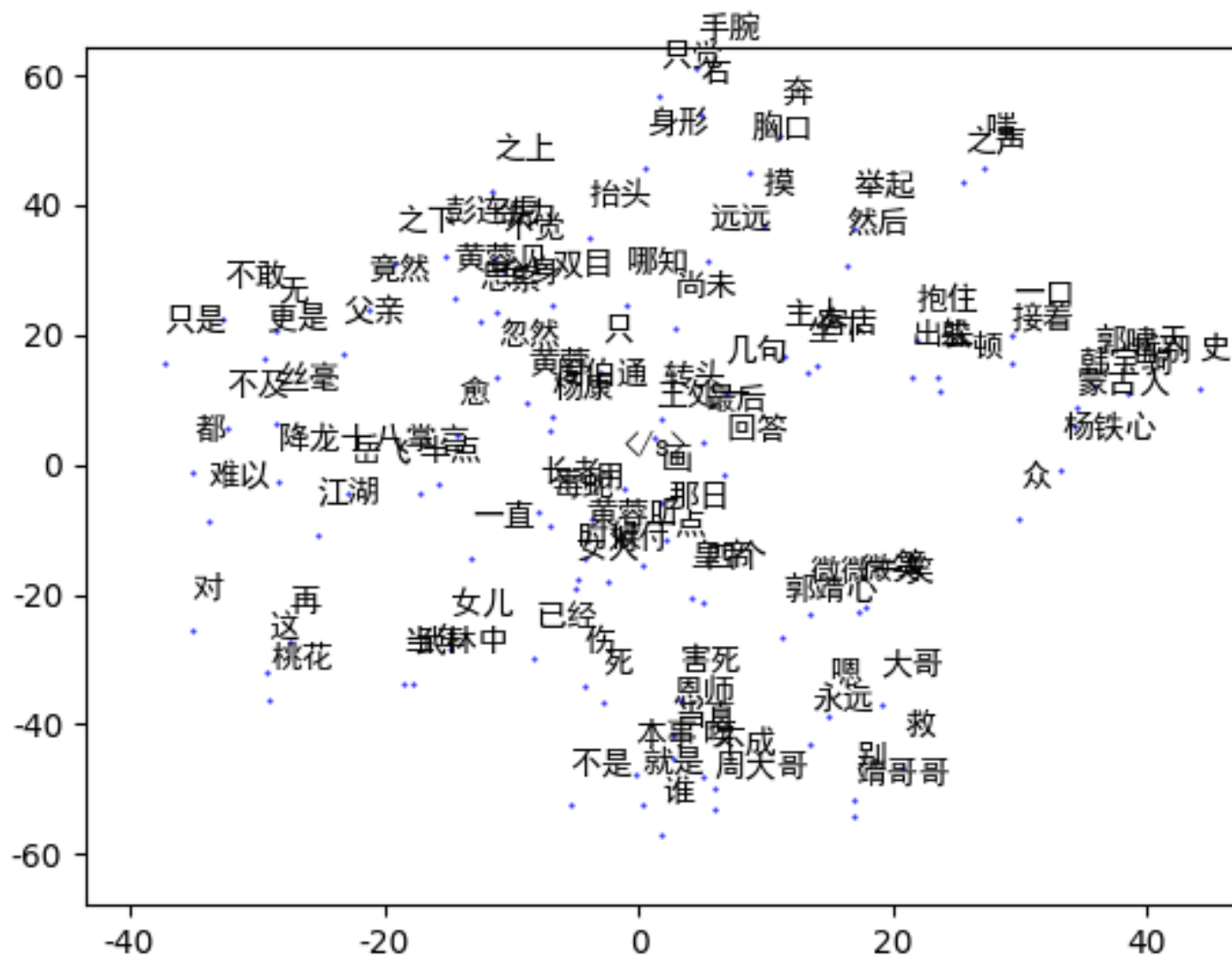
#STEP2: generate word vector
/tools/srilm/ngram-count -text seg.txt -order 1 | sort -k2 | awk '{if (int($2) > 10) {print $1}}' | grep -v [ :"., ' ? \ \. "\(\)\*\*\<>> ; + | grep -v '<s>'
| grep -v '<s>' > dict.txt
/tools/word2vec/word2vec -train seg.txt -output wordv -read-vocab dict.txt
awk '{if (NF >= 2){$1=""; print $0;}}' wordv >word_vec.txt
awk '{if (NF >= 2){print $1;}}' wordv >word_name.txt

#STEP3: tSNE to draw the vector
python draw.py word_vec.txt word_name.txt wordv.pdf

evince wordv.pdf &
```



# 修改词频后



# 实验二：生成人名的词向量

- 绘制《射雕英雄传》中人名的词向量空间，观察人物之间的关系
- 对词表 dict.txt 进行筛选，保留人名存至 name\_list.txt 中
- 在词向量文件 wordv 中筛选出对应人名的词向量
- 利用 t-SNE 工具，将人名词向量投影到二维空间

# 实验步骤

- 本次实验我们单独选出《射雕英雄传》中部分人物，保存到 name\_list.txt
- 运行 run-name.sh，查看他们之间的词向量的关系距离。

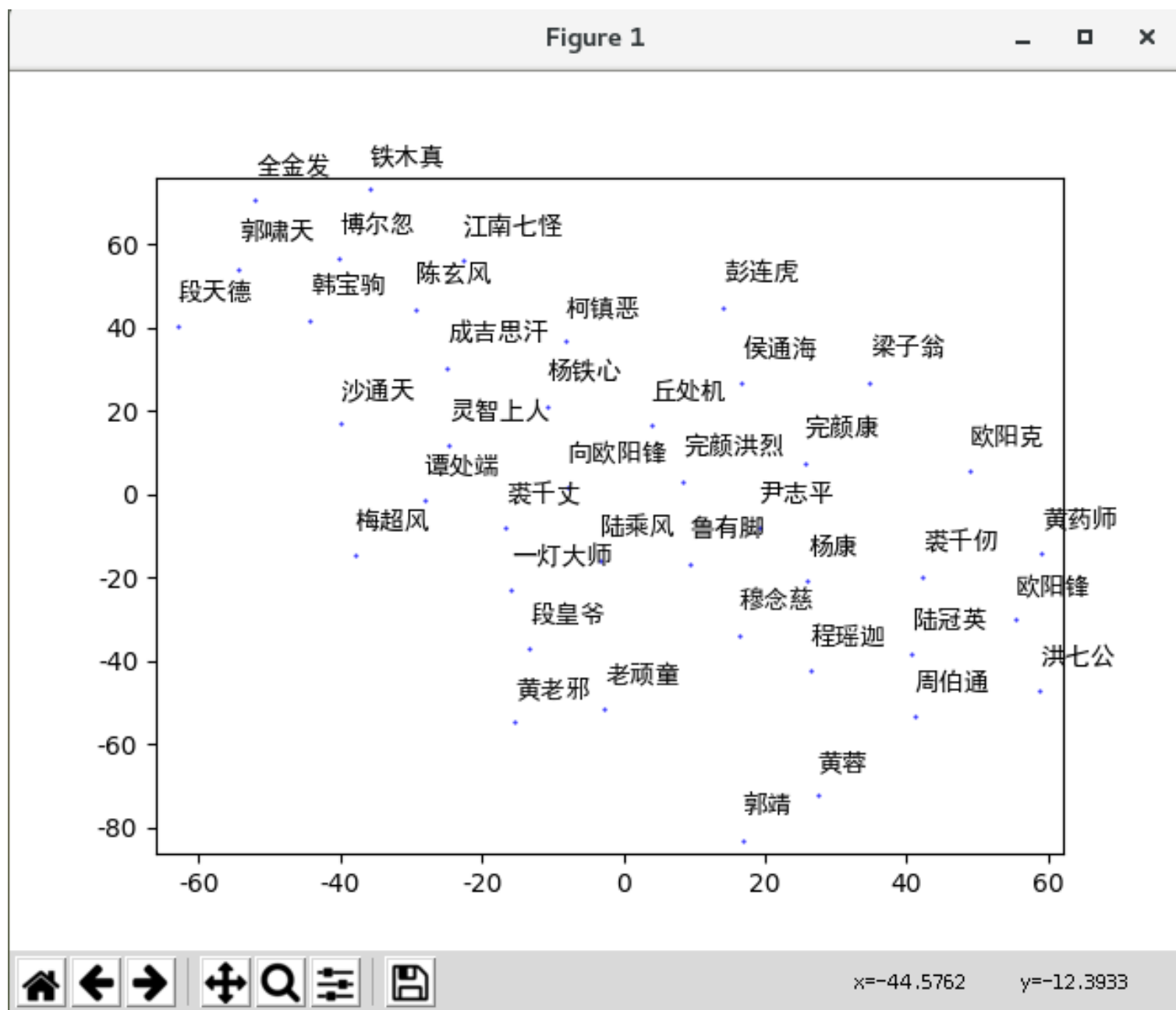
## sh run-name.sh

```
centos7 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
应用程序 位置 终端
tutorial@localhost:~/aibook/demo/lang/word2vec/code
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[tutorial@localhost ~]$ ls
aibook 公共 模板 视频 图片 文档 下载 音乐 桌面
[tutorial@localhost ~]$ cd aibook/demo/lang/word2vec/code/
[tutorial@localhost code]$ ls
dict.txt  name_list.txt  run.sh  seg.txt  word_name_name.txt  wordv  word_vec.txt  wordv_name.pdf
draw.py  run_name.sh  seg.py  txt  word_name.txt  word_vec_name.txt  wordv_name  wordv.pdf
[tutorial@localhost code]$ sh run-name.sh
```

路径

运行

# 实验结果





**The end**