## 基于复杂网络的英语单词记忆系统的设计与实现

## Functions and Thoughts

2019.10.18起

<https://blog.csdn.net/weixin_41803874/article/details/92845206>

（关于研究内容如何写）

<https://zhuanlan.zhihu.com/p/80682302> （编辑距离-面试题）

有问题务必及时联系导师！

### Set up and Model the graph based on the data

1. 找单词库
2. 建模（图）：

词义（中文意思）相近相反[近义词&反义词（除了加前后缀&调用词库）？？]

词形（词根词缀）相近[或eg. Apple cripple]

相似词（eg.）：用编辑距离（替换、插入、删除）来衡量

词组（词组库），

如何实现分类？？

以不同权重来区分

词向量？?将词化为稠密向量，相似词的词向量相近

一个n个字母组成的单词，有n-2个3字母(相邻)组合，有n-3个4字母(相邻)组合

把这些组合做标记(vector)，

例如:apple有app，ppl，ple的字母组合；也有appl，pple的字母组合

Cripple有cri，rip，ipp，ple，ppl；也有crip，ripp，ippl，pple；又有cripp，rippl，ipple；和crippl，ripple

词形变化

1. 计算并分析和寻找特征（最小编辑距离，平均距离），画边，加权重
2. 设计数据库，并存入数据库

### Regarding Theory

#### 编辑距离

动态规划，可衡量两个字符之间的相似度（庞大的数据怎么处理？？先用Shingling去除很不相似的单词，再用）

Shingling to remove less-alike words: Base on K=2, K=3, K=4, K=5，K=6（看单词的情况具体分析，一般6就足够了，例如subsequence中quence）K的值越大，剩下的单词相似度越高（很短的单词，比如三个字母，那就K=2即可）

(edit distance also The [Levenshtein distance](https://en.wikipedia.org/wiki/Levenshtein_distance" \o "Levenshtein distance)):Frequently used in NLP, especially for automatic word correction. Replace a word with a lower ED in question.

Variants：[Longest common subsequence (LCS)](https://en.wikipedia.org/wiki/Longest_common_subsequence_problem" \o "Longest common subsequence problem)[insertion and deletion]; , [Hamming distance](https://en.wikipedia.org/wiki/Hamming_distance" \o "Hamming distance)[substitution, same length of word]

Common prefixes and suffixes can be skipped in linear time

### Mini-Program

dsf

### Paperwork （20 pages++）