第一次作业--词汇相似度计算

背景介绍

> 计算词汇相似度，并基于Mturk-771进行实验和分析(开放式)：
>
> 1. **基于语义词典的方法(Thesaurus-based)**
>   基于两个词在WordNet等语义词典中是否“相邻”
> 2. **基于语料统计的方法(Distributional/Statistical algorithms)**
>   比较词语在语料库中的上下文

工具和数据集

实验数据：

- Mturk-771

环境：

- Win10
- Python 3.6.5

工具：

- nltk (wordnet)
- gensim (word2vec)
- scipy (spearman's)

训练数据：

- word2vec Text8 (Wikipedia)

评价方法：

- Spearman's rank correlation coefficient

算法

WordSimilarity中实现了 **7** 种词汇相似度计算算法，分别为：

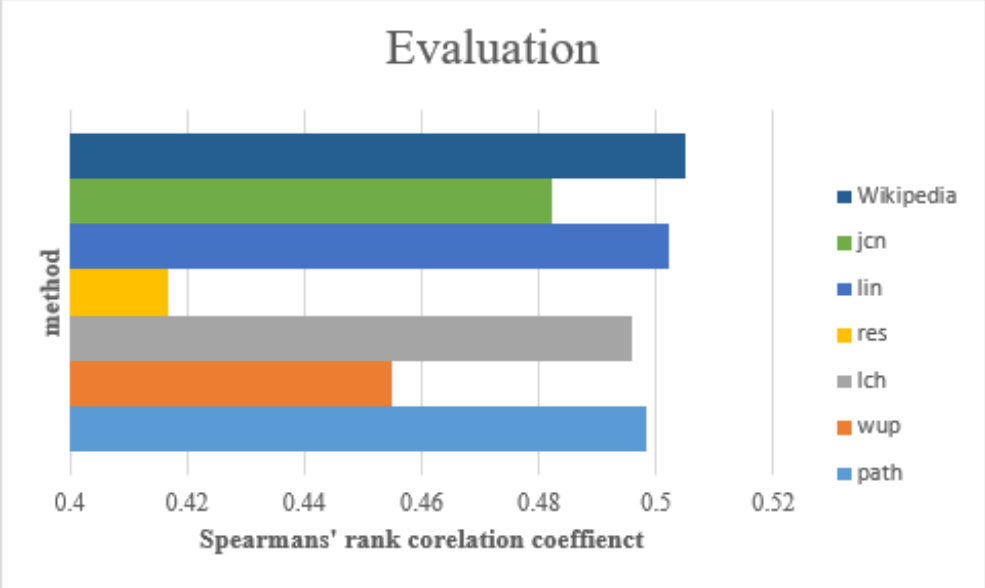
- **基于WordNet的方法（包括路径、互信息）**
  - wup
  - path
  - lch
  - res
  - lin
  - jcn
- **基于语料统计（Wikipedia）的方法**
  - word2vec (text8)

使用方法

```
python main.py
```

实验结果

Type	Method	MTURK-771
WordNet	path	0.498492488292959
WordNet	wup	0.455005384563715
WordNet	lch	0.496041905015696
WordNet	res	0.416802424371286
WordNet	lin	0.502236568195814
WordNet	jcn	0.482319181190982
Word2Vec	Wikipedia	0.505088289124088



分析与结论

1. 利用Wikipedia语料的方法明略优于基于WordNet的方法。其原因在于WordNet的信息量比较有限，一些信息内容文件没有词性条目：`Information content file has no entries for part-of-speech`，而且收录的词语不同词性之间也无法计算语义相似度。
2. 6种不同的基于WordNet的算法中，Resnik method表现最为糟糕，原因是该方法只考虑了两个词的共性信息，即计算Sim时只利用了LCS而并未充分挖掘单词各自的内容信息。

3. 在实验中，用于训练的Wikipedia语料仍然有限(Text8大小约为100MB)，一些词语未被收录：`word2vec: "word 'washer' not in vocabulary"`，因此如果使用更多语料，可能会获得更好的结果。

致谢

本报告只用于课程sckr2019@PKU，部分内容参考自GarD0u的博客。