北京大学

研究生课程作业报告

题目 **Task1 词汇相似度计算**

姓　名：姜慧强
学　号：1801210840
院　系：软件与微电子学院
专　业：软件工程
研究方向：智能化软件

二0一九年三月

一、基于词典

基于词典的方法，基本思路是通过搜索词典中关于单词的解释。如果解释中存在另外一个词语，那么认为这两个单词相关度较高。

- **wordNet**

使用 wordNet 作为词典。使用 nltk 的 wordNet API 进行相似度计算。

利用 wn.synsets(word1)选出词典中 word1 所有解释，按照每个解释计算相应的相似度。

在计算过程中，就筛选词典中解释的过程做了一些对比试验。

Word1 在词典中有 k1 个解释，word2 在词典中有 k2 个解释。总共就有 k1 • k2 个 pair 对。如何用着 k1 • k2 个 pair 对衡量最后的 result。可以是取最大值，可以是取均值。

另外，可以利用 lemma_names() 提取出词典中关于 word 的注释中对应的单词。那么 word1 可以提取成为一个长度为 m1 的 list，word2 可以提取成为长度为 m2 的 list。这样再分别对 m1 • m2 对分别做上述操作。这样相对于可以做两层 wordnet。然后也分别求得最大值，平均值。

评价指标选用 spearmanr 指标。

（one）(two)	*spearmanrResult*
Basic Max	0.4994
Basic Mean	0.4193
Two Max Max	0.4255
Two Max Mean	0.5016
Two Mean Max	0.3418
Two Mean Mean	0.2813

可以看出，对词典注释计算得到的结果取 max 普遍比取 mean 效果要好。个人感觉这样的结果也比较符合人类一般认知。在人类对词语和词语之间相关度判断的时

候，实际上也是取一个词条意思相关度的最大值。从大脑的认知中，搜索 word 对应的含义，当这两个词有某个含义相近的时候，我们就认为他们是相似的词，而不会去对每个词的含义做一个平均。而当我们做两轮 wordnet 时候，如果再取 max 的话，词义偏差就会被放大，这个时候反而会出现效果下降的情况。所以综上可得，当利用 wordNet 做词汇间相似度计算的时候，对所有解释 pair 的相似度取最大值效果更符合人类认知，也有更好的实际效果。

二、基于语料

和基于词典的方法，不同的是，基于语料方法，基本思路就是就是 word2vec 的思路。从周围词推中心词，从中心词推周围词，再加上 negative sample 和 Hierarchical Softmax。然后把 NNLM 的隐藏层作为 word 的 Embedding。此后，也有类似 ELMO 基于上下文的 word Embedding 被提出。通过计算每个词的 Embedding，通过欧氏距离判断词汇的相似度。

1. word2vec

根据 Wikipedia 的语料训练 skip-gram 模型，skip gram 是利用周围词预测中心词。从而得到语料库中所有词汇的 word Embedding 值。然后根据这些 word Embedding 值对词汇进行余弦距离计算。再通过 spearmanr 进行评价。

除了直接基于 word2vec 训练得到 Embedding 进行点乘计算相似度之外，还利用 wordNet 拓宽单个词汇至临近的几个词。

	spearmanrResult
Basic word2vec	0.4602
wordNet Max	0.4360
wordNet Mean	0.5214
wordNet Median	0.3829

从上文可以看出 word2vec basic 的效果相较于 wordNet 稍微弱一点。在加上 wordNet 的词语扩展之后，通过 mean 处理的 similarity 提升相似度还是比较明显的。符合之前得出的二重 wordNet 使用均值可以减弱因为词义偏离造成的性能负效应。

2. bert-Embedding

以 bert 官方提供的训练模型为基础，利用 gluonnlp 库对训练好的模型提取出 Embedding 值，其余同上。和 Word2vec 一样，对 bert 也引入了 wordNet 拓宽词汇处理。

		Basic	*wordNet Max*	*wordNet Mean*
bert_12_768_12	book_corpus_wiki_en_uncased	0.2234	0.1879	0.2249
	book_corpus_wiki_en_cased	0.1620	0.0754	0.2316
bert_24_1024_16	book_corpus_wiki_en_uncased	0.2470	0.2024	0.1863
	book_corpus_wiki_en_cased	0.1549	0.0308	0.2263

Bert-Embedding 的效果反而没有预想的那么好，使用官方提供的几个训练模型做 Embedding 之后，计算相似度，spearmanr 指标有些低，可能在处理的时候出了一些问题，但截至提高报告时还未查出具体出在哪。

但抛开整体偏低的指标，可以发现几次实验中 WordNet 的 Mean 大多相较于 Basic 有所提升。WordNet 的 Max 大多相较于 basic 有所下降，基本符合前面做出的推断。

3. googlenews

最后利用 googlenews 已训练好的 word Embedding 做了一个词汇相似度的对比试验。其中利用 gensim 导入已经训练好的 model。同样也用 wordNet 做了一个拓宽词汇的处理。

	spearmanrResult
Basic GoogleNews	0.6710
wordNet Max	0.5076
wordNet Mean	0.5435
wordNet Median	0.4599

可以看出 GoogleNews 在 MTURK-771 上效果较好。如果加上 wordNet 的词关系，反而会掉点。但如果只看加上 wordNet 之后几个指标的处理，可以发现，结论与

前面一致，即在拓宽词汇过程中，使用均值可以减轻因为拓宽词汇造成的词义偏离。最大值和中位数，相较于均值效果更差。

三、总结

本次实验，从基于词典、基于语料两个方面出发在 MTURK-771 数据上验证了词汇相似度计算的效果好坏。模型的评判使用 spearmanr 指标。

其中在基于词典的方法中，采用了二重词典，并得出一重使用最值，二重使用均值效果最好。二重效果比一重更好的结论。

在基于语料的方法中，选用了 word2vec，bert-Embedding，googleNews 三种方法。其中 bert-Embedding 和 googleNews 两种方案使用的是已经训练好的模型，word2vec 是基于手写的 skip-gram 实现的。并在这三种方案中，分别利用 wordNet 进行词义的拓宽对比试验。根据实验结果可以得出，部分模型下，利用 wordNet 拓宽词汇时使用 mean 方式做多词对的 similarity 做采样效果会好于 basic 模型。实验过程中 googleNews 的模型效果最佳。