词汇相关度计算实验报告

引言

本文实现了 2 类词汇相关度的计算方法，分别基于词典和基于语料。本文在 Mturk-771 数据集[1]上进行了实验和分析。

Mturk-771 数据集包含 771 个词对，词对相似度由人工标注，并做了数据处理，相似度值在 1 到 5 之间。本文采用 Spearman 秩相关系数[2]来评价词汇相关度计算方法的有效性。本实验中，Spearman 秩相关系数用于衡量人工评价的相似度值与计算方法得出的相似度值之间的相关性，相关系数越大，则相关度越高。

基于词典的词汇相关度计算方法

本文尝试了 4 种基于语义字典的词汇相关度计算方法，代码实现上采用了 nltk 中的 wordnet 库[3]。Path Similarity(path)、Leacock-Chodorow Similarity(lch)是通过路径距离来计算词汇的相关度。Wu-Palmer Similarity (wup)是利用两个词节点的最低共同祖先节点等拓扑信息来衡量词汇相关度。Lin Similarity(lin)改进了 Wu-Palmer Similarity，通过额外的语料库来估计概念节点的概率，从而赋予了 wordnet 结构概率信息。实验中，采用 nltk 的 ic-brown 作为额外的预料库。由于每个词可能有多个词义单元，故将两个词义单元间的相似度的最大值或平均值作为词汇的相关度。通过计算 Spearman 秩相关系数来评价词汇相关度计算方法的有效性，实验结果如表 1 所示。

	max	average
path	0.498	0.411
lch	0.496	0.309
wup	0.455	0.263
lin	0.493	0.345

表 1 基于 wordnet 的词汇相关度计算方法的实验结果

实验表明，使用词义单元之间的相似度的最大值作为对应词汇的相似度，其效果要好于使用平均值。在 Mturk-771 数据集上，Path Similarity 的效果最好。通过引入了概率信息，Lin Similarity 的效果优于 Wu-Palmer Similarity。

基于语料的词汇相关度计算方法

使用 word2vec 计算词对应的稠密表示，并将词向量之间的 cosine 相似度作为词汇相关度。代码实现上采用了 gensim 库。

实验参数设置:词向量维度设定为 300，语料采用 text8[4]，epochs 设置为 10。实验结果如表 2 所示，Spearman 秩相关系数为 0.540，效果要显著好于基于 wordnet 的方法。

---

[1] http://www2.mta.ac.il/~gideon/mturk771.html

[2] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

[3] http://www.nltk.org/howto/wordnet.html

[4] http://mattmahoney.net/dc/text8.zip

考虑到 text8 语料库规模仅有 100MB，且可能存在 word2vec 训练不充分的情况，本文采用谷歌发布的词向量[5]进行了实验。该词向量的语料主要来源于新闻领域，词向量维度为 300。实验结果如表 2 所示，使用谷歌预训练的词向量，Spearman 秩相关系数高达 0.671，相较于通过 text8 语料训练，效果提升了 24.3%。

基于语料的方法可捕捉词汇上下文之间的共现关系，信息损失要少于基于 wordnet 的方法，效果显著优于后者。基于语料的词汇相似度计算方法的效果与语料的规模及语料的分布领域有较大相关性。

	Spearman's rho
train from scratch	0.540
pre-train	0.671

表 2 基于 word2vec 的词汇相关度计算方法的实验结果

总结

如表 3 所示，基于 word2vec 的词汇相关度计算方法效果要显著优于基于 wordnet 的方法。前者可捕捉词汇上下文之间的共现关系，信息损失要少于后者。基于语料的词汇相关度计算方法，精度高于基于字典的方法；但其在训练过程中所需的计算资源较多，效率低于基于字典的方法。

	Spearman's rho
wordnet	0.498
word2vec	0.671

表 3 实验结果总结

[5] https://code.google.com/archive/p/word2vec