09017232刘晓臻\_课程研学报告（二）

**学号：09017232 姓名：刘晓臻 日期：2020.04.19**

# 一、构建查询需求

报告（一）中已经指出，原文档标题“Effective Online Knowledge Graph Fusion”作为查询，即使是在其本身的（35篇）文档库中，有关联的文档也仅3篇。经过在学术搜索引擎中的查找，哪怕以排序结果的前10篇（其中有7篇无关）作为理想文档集，对标题中的词排列组合，也无法在Google Scholar搜索的前100名结果中找到理想文档集里有的内容！

因此，不能使用原标题和原文档库，否则本实验无法进行，因为最终各项评分结果都是0！

报告（一）中还给出了另外一篇论文（原论文的引文）《Computing semantic relatedness using wikipedia-based explicit semantic analysis》，以及用它构建文档库和查询的结果，在此使用本篇论文的标题构建查询（能够在后续搜索中搜索到理想文档集里有的）如下（<queries.txt>）：

semantic relatedness

computing semantic relatedness

semantic relatedness semantic analysis

wikipedia semantic relatedness

semantic relatedness wikipedia based

semantic relatedness wikipedia based semantic analysis

relatedness wikipedia based

# 二、构建理想相关文档集

根据报告（一）中得到的结果，选用向量模型的top10文档作为理想相关文档集<ideal_docs.txt>如下：

WikiRelate! Computing semantic relatedness using Wikipedia

Indexing by latent semantic analysis

Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge

Evaluating wordnet-based measures of lexical semantic relatedness

Extended gloss overlaps as a measure of semantic relatedness

Exploring unexplored contexts for semantic extraction from syntactic analysis

Centroid-based document classification: Analysis and experimental results

Corpus-based and knowledge-based measures of text semantic similarity

Feature generation for text categorization using world knowledge

Feature Generation for Textual Information Retrieval Using World Knowledge

# 三、选择搜索系统

选择Google Scholar作为搜索系统，对每个查询取前100个标题，得到查询结果在<query_results>文件夹下。”

# 四、评价查询的检索效果

### 1. 每个查询结果集

通过代码<evaluation.py>对每个查询结果集计算得到Precision-Recall对应关系及整个结果集的Precision，Recall及如<evaluation_results.txt>所示：

semantic relatedness:

{0.1: 0.5, 0.2: 0.6666666666666666, 0.3: 0.75, 0.4: 0.05333333333333334},

recall: 0.4

precision: 0.04

computing semantic relatedness:

{0.1: 0.5, 0.2: 0.06896551724137931},

recall: 0.2

precision: 0.02

semantic relatedness semantic analysis:

{0.1: 0.06666666666666667, 0.2: 0.06060606060606061},

recall: 0.2

precision: 0.02

wikipedia semantic relatedness:

{0.1: 0.5, 0.2: 0.02247191011235955},

recall: 0.2

precision: 0.02

semantic relatedness wikipedia based:

{0.1: 0.16666666666666666, 0.2: 0.04081632653061224},

recall: 0.2

precision: 0.02

semantic relatedness wikipedia based semantic analysis:

{0.1: 0.041666666666666664, 0.2: 0.029411764705882353},

recall: 0.2

precision: 0.02

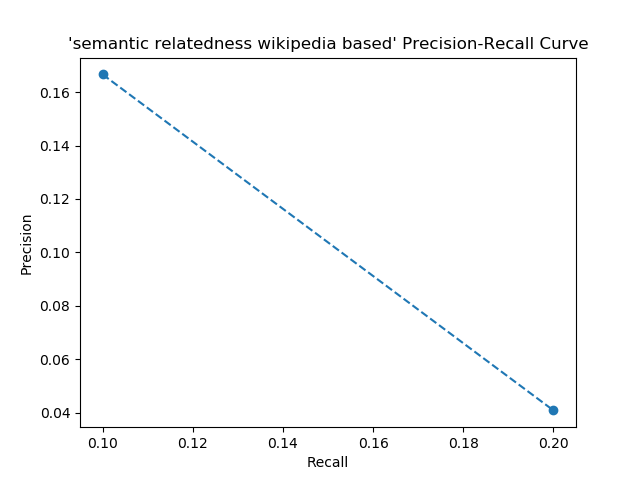
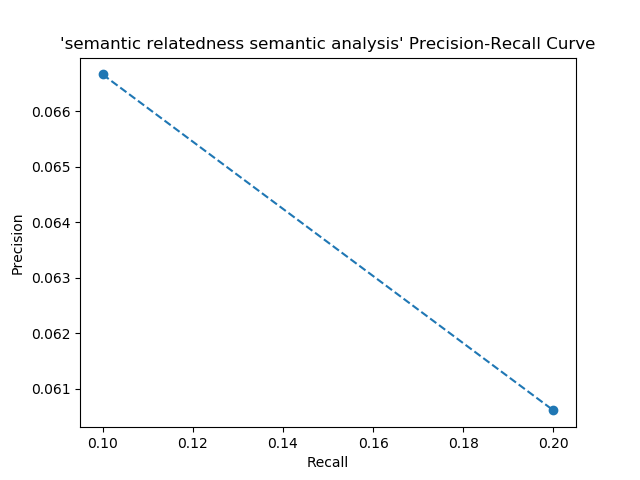
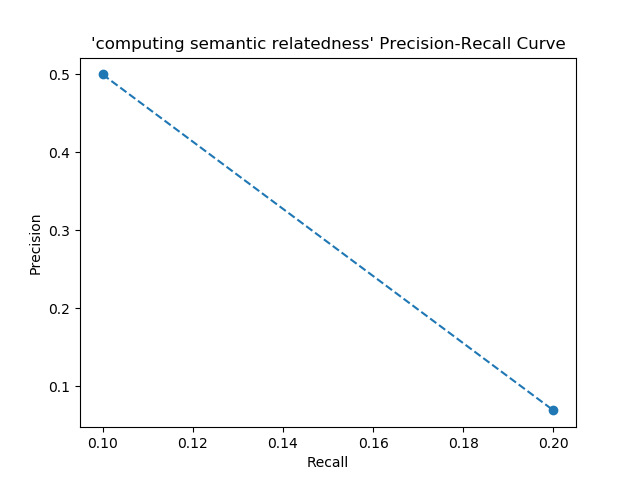
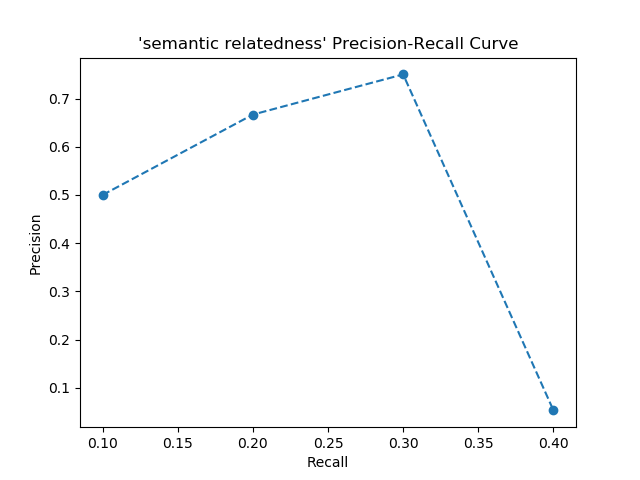
relatedness wikipedia based:

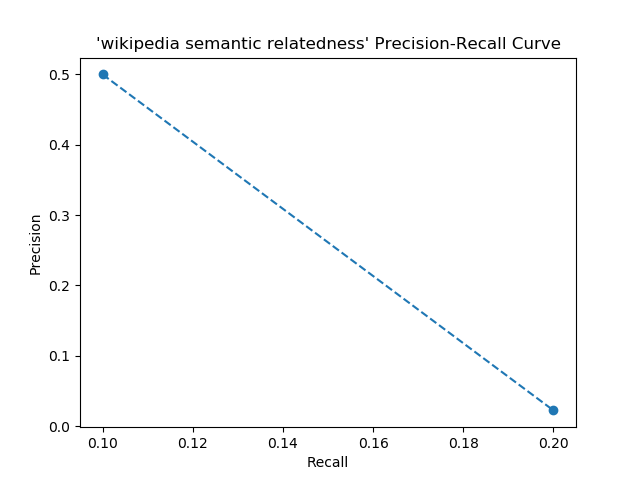
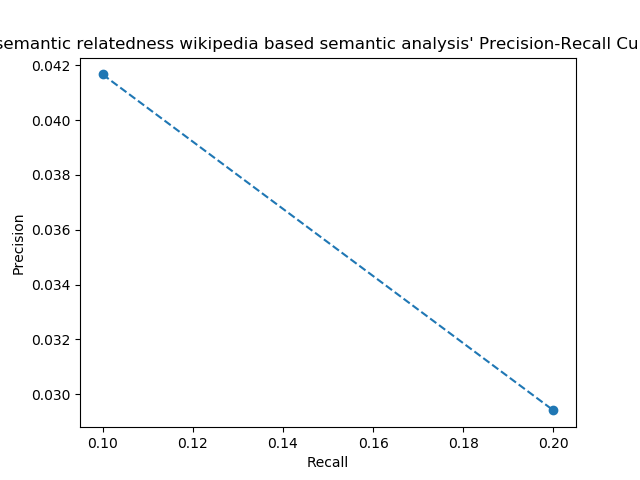
{0.1: 0.5, 0.2: 0.038461538461538464},

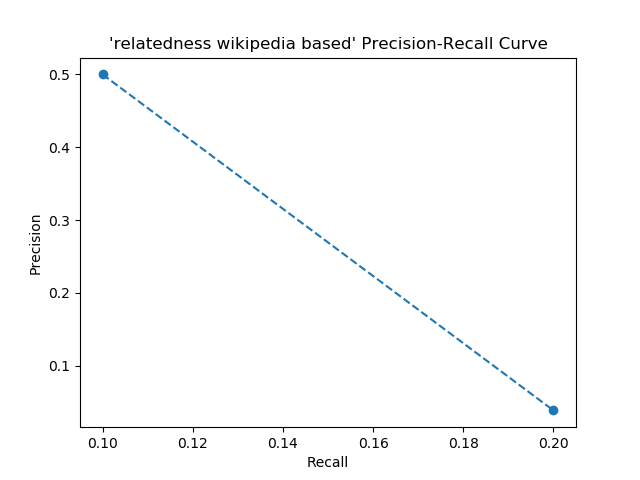
recall: 0.2

precision: 0.02

通过以上P-R对应关系，可以画出每个查询的P-R曲线如下（对应文件在<figures>文件夹里）：







### 2. 平均效果

同样使用代码<evaluation.py>里，计算出对于q的所有查询结果集的平均Precision为0.022857142857142857，对于每个Recall值的平均Precision结果输出在<evaluation_results.txt>里。综上，可以画出平均Precision-Recall曲线<figures/avg_pr.png>，如下图。

