- **why is this paper named “topic-sensitive”?**
  因为这篇文章中的搜索方法，为了得到更加精确的结果，关注到了询问词（query）本身，也关注到了网页的话题属性。在预先计算的一系列具有代表性的话题对应的 PageRank 向量的基础上，通过对询问词所属的话题的 PageRank 向量进行线性组合的计算，来计算在特定话题背景中各个网页的“话题敏感”的评分，从而得到与询问词相关的“话题敏感”的网页。

- **What’s the problem this paper was trying to address?**
  传统的 PageRank 算法在线下一次计算所有的页面的重要度，仅仅利用了网页的链接结构来判断返回网页的排序，更加关注网页的“重要程度”，与查询过程是独立的，忽略了询问词本身。由于一些网页在某些话题下很重要，而在其他话题则不然，这些被认为“很重要”而被的网页有可能并不能与询问词的主题很好地契合。

- **What’s the role of matrix D?**
  注意到$D = p \times D^T$，其中D的作用是指示某一个网页是否出度为 0，若为 0 则对应项为 1。其含义是，这些出度为 0 的网页将按照p中指示的概率访问其他网页。公式中的“概率矩阵”为$(M + D)$，其中M矩阵揭示了从网页链接结构中得到的概率关系，当某一网站出度为 0 时，其对应的列全为 0；若考虑D矩阵，实际上综合考虑了“网页的链接结构”和“没有出度的网页”两种情况。这也就对应了在随机游走模型中，当一个网页出度为 0 时的访问页面的情况。
  进一步观察，当访问到任意页面时，实际上都考虑了“按照链接结构访问”和“按照p随机访问”两种访问情况，并分别设置总和为 1 的权重。对于没有出度的页面，实际上仅能进行“按p访问”；增加 D 矩阵后，两种访问情况得到统一，且权重和为 1，更加切合真实情况。
  在后文中，由于每个类别c_j都有一个单独的p_j，因而也有对应的若干个D_j按照相同的办法得到。

- **Why we need the ODP or other similar project for the computing of multiple page ranks for a given page?**
  因为将传统的 PageRank 算法修改成话题敏感的、考虑更多的 PageRank 算法，需要对 Rank 计算公式增加一些“偏向”；首先要做的，是提前利用爬虫得到一组对应于各个话题的 PageRank 向量，而这需要一个具有“基准”话题集合以及各个话题对应的网页的集合。基于这个目的，文章选择了 ODP 这样一个免费、实用且可靠的信息源来构建向量。