Brief Article

The Author

October 23, 2013

1 Abstract

2 Introduction

Ad-hoc搜索的任务是从一堆静态的网页文件中搜索出与给定查询最相近的文档。今年的web track使用的是clue2013数据集。Web clue 是由cmu提供的xxxx抓取了一年的数据。我们的实验采用的是clueB数据集，它是clue2013的一个子集，约500G。本文主要分为两个部分，第二章主要是介绍索引的建立，第三章是基于查询扩展的搜索模型，第四章是实验结果分析，最后一张是总结和展望。

3 Build Index

我们适用了Indri工具作为索引构造器，配合spammer [?]，对数据进行建索引。

4 Search

在搜索策略中，我们采用了查询扩展的方法。扩展词是利用元搜索的搜索结果聚集在一起，经过分析处理之后，得到全中最大的词，然后再经过词性分析，保留了名词。第二步是构造新的查询语句，这里我们使用了查询扩展的部分结果，在查询扩展中与查询词中相同的词的词频作为归一化分母，从而得到了每一个扩展词的权重，进而构造出对应的查询扩展语句。最后一步则是对搜索结果进行reranking。

1.查询词扩展

在查询扩展的数据来源中，我们分两个扩展来源进行，一个是使用了元搜索的方法，利用api调用的方法把google search和bing的搜索接口统一调用，对50个topics分别进行搜索并返回其前50项搜索结果。然后使用了基于Block的正文抽取算法[?], 得到各个网页的正文内容，然后根据TF的数量进行排序，从而得到初始的扩展词表；另一个是对wikipedia中相关的页面进行锚文本的计算，通过各个锚文本链接的相互关系，使用pageRank的方法得到权重最高的锚文本节点。 第二步，我们使用了stanford的nlp分词软件，将这些词进行词根化和词性标注，仅保留名词作为扩展词。

2.构建查询语句

我们将扩展词中与原查询的同义词或者本身词作为最大项，然后对其他词的词频进行处理得到权重 $w_i = \frac{TF_i}{TF_{max}}$,然后扩展原查询 $q_{expan} = q_{origin} + \sum_{i=1}^{n} w_i * Expan_i$, 将此扩展后的查询作为新的查询。

3.重排序

利用扩展后的查询放到Indri中进行搜索，主要使用BM25语言模型。

5 Experiments

本次由于我们只提交了一个run，在B数据集上的结果效果很不好。

6 Conclusion Future

通过这次实验，我们验证了查询扩展在辅助查询的方面有所帮助。

6.1 A subsection

More text.

References