基于语义关系查询扩展的文档重构方法*

张　敏，宋睿华，马少平

清华大学计算机科学与技术系 智能技术与系统国家重点实验室，中国北京　100084;

E-mail:　zhangmin@s1000e.cs.tsinghua.edu.cn

摘要

已知文档与用户查询之间相同概念不同表达形式造成的词不匹配问题，是影响信息检索效果的重要原因之一。本文提出了根据词之间的语义关系进行扩展和替换的文档重构方法。它与传统的查询扩展不同，实现了同一概念信息的聚集，是更接近于人类进行信息查找的思维方法。进一步地，研究给出一种有效的实时文档重构检索策略，解决了文档重构方法在实际应用中的可行性。在标准测试数据集上的实验表明，基于查询扩展的文档重构方法不仅比不扩展的最佳性能相比始终有 14%到 23.4%的提高，而且比相对应的传统查询扩展方法也有约 16%的提高。

关键词 文档重构 查询扩展 语义相似性 词不匹配 概念检索

Document Refinement Based On Semantic Query Expansion

ZHANG Min, SONG Ruihua, MA Shaoping

State Key Lab. of Intelligent Tech. and Sys., CST Dept. , Tsinghua Uni., Beijing 100084, China;

E-mail: zhangmin@s1000e.cs.tsinghua.edu.cn

Abstract The word mismatch problem of various expressions of the same concept between known document resources and user query is one of the main factors that hurt the retrieval performance. This paper proposes a document refinement (DR) approach by expansion and replacement based on semantic relations between words. Different from traditional query expansion technology, this DR approach clusters the information with the same concept, which is closer to human thinking habit of information seeking. Furthermore, an effective real-time DR strategy has been given, which makes the DR approach feasible to applications. Experiments on standard testing set showed that the DR approach made consistent 14% to 23.4% improvements after refinement, and got 16% improvements compared with the corresponding query expansion technologies.

Keywords document refinement, query expansion, semantic similarity, word mismatch, concept retrieval

一．背景及相关研究工作

在当前的信息检索模型和系统中，信息都是以字、词或者词组的形式来表示的。只有查询词出现在文档中时，才有可能被检索到。但是在自然语言里同一个概念经常会有多种不同的表达方式，因而很有可能出现与用户查询含义相关的文档由于用词不同而无法被检索出来的情况。这种词不匹配问题是影响信息检索效果的重要原因之一。

*本项目得到国家自然科学基金资助（60223004, 60321002, 60303005）

例如，对于 automobile recall（汽车回收）（选自 TREC2002 Novelty Track Topic397）这样的用户查询，可能相关信息在文档中以如下的方式表达：

1．*FORD* is recalling 57,000 Mondeo family saloon *cars* in the UK because of a minor defect to the handbrake lever.

2．*Mazda* Announces Recall: Mazda Motor Corp. announced the recall of 102,548 mid-sized 626 *sedans* from the 1986- and 1987-model years to fix a faulty ignition switch.

3．*Ford* is recalling 57,000 Mondeo saloons in Britain because of a small defect in the *handbrake* lever.

其中三个句子中，都没有出现查询词 automobile，但是三个句子都和用户查询直接相关。其中句 1 中的 car 是 automobile 的同义词；句 2 中提到的 sedan 通常指小轿车，在语言学上是 automobile 的下义词；句 3 与句 1 表达的意思完全相同，其中 Ford、handbrake 等词也是与 automobile 相关的内容。另外，句 1 和句 3 在表达英国时，一个使用了 Britain 而另一个使用了 UK，也是同一个概念有不同表达方式的一个例子。

由于词不匹配问题的存在，用户有时不得不变换查询词才能找到所需要的信息。减轻这种用户负担的一种方法是由检索系统自动选择一些与查询词相关的其他词项来辅助查询,即查询扩展技术。简单的说来，查询扩展就是检索系统在进行检索之前，先根据扩展词表，自动把用户查询中的关键词的同义或者近义词扩展进来形成新的查询，然后再进行检索。

容易知道，查询扩展中最关键的技术之一就在于扩展词表的构造。目前扩展词表的构造通常有三种方式：第一种是根据语言学知识基于语义的查询扩展词表构造方法[1][2][3]，并构建了一些大规模的手工词典例如 WordNet[4]、HowNet[5]等；第二种是基于大规模通用语料库的统计信息例如同现概率、互信息等构造扩展词表[6][7][8][9][10]；第三种是结合语言知识和统计信息的扩展词表构造方法[11][12][13]，例如基于依存关系统计信息的扩展词表[14][15]。

在基于语义的查询扩展研究中，人们经常利用 WordNet 里提供的同义词集合和 is-a 关系（上/下义关系）来选取新词扩展查询。但是从查询词出发扩展多层下义词时，扩展词的数量会随着层数的增加而快速增长，同时扩展词中无用词的数量也极大增加。因此扩展的层数确定是一个尚未解决的问题。怎样使用扩展词也是一个问题，一般认为原始查询词最能反映用户的需要，而扩展词的准确性值得怀疑，因此在使用扩展后的查询时会对原始查询词赋予较高的权重，对扩展词赋予较低的权重。但是究竟应该设为什么权值则一直没有很好的方法，通常依靠经验值给出。Voorhees[16]尝试了各种权重，甚至手工的挑选扩展词，只是得到不超过 2%的性能提高。

本文考察人类信息检索的思维方式，与传统方法进行比较，提出一种与查询扩展思路相反的文档重构思想，以解决检索中的词不匹配问题，同时避免了查询扩展中的权值设定问题。论文的第二节介绍基于查询扩展的文档重构思想及基本算法；第三节对传统的查询扩展方法和文档重构进行比较分析；第四节提出一种实时文档重构算法，解决了该方法的实际应用问题；第五节给出实验结果及分析；最后对本研究工作进行总结。

二．基于查询扩展的文档重构

传统的检索查询扩展的方法，都基于这样的思路：把文档做为已知内容，而在检索中对查询进行扩展。即从查询的角度进行信息的修正然后加以匹配。这种方法在处理上具有简单易行的优点。

但是考察人类进行信息检索的行为，发现人们自身的检索思路完全不同：在人类进行信息查找时，查询是一个已知信息，而在看到每一篇文档时会根据查询对文档中的内容进行修正，以寻求信息的匹配。这与传统的查询扩展方法在思路是相逆的。进一步地，因为文档内容的修正是根据查询词表示的概念进行的，因此可以认为是一个对查询进行扩展和对文档进行修正的双向过程。

比较两种不同的思路，我们发现二者最大的区别在于信息的发散和聚集。我们可以把查询中的

每一个查询词看作一个信息单元或独立的信息概念。传统的查询扩展方法是把这些概念具体化，并在检索中把这些实际上具有密切语义联系的、属于相同概念的词语看作独立的单元来处理。对于那些概念上较抽象而其包含的信息内容丰富、概念层次复杂的查询来说，这种扩展方法很难找到合适的扩展层次。相反地，在人类的检索过程中，是把文档中那些具有密切语义联系的词语全部纳入一个概念中来，是一个信息聚合和加强的过程，也就不存在聚合层次及扩展词权值难以确定的问题。

基于这样的考察，从人类思维过程中得到启发，我们提出一种基于语义关系查询扩展的文档重构方法，从而实现信息的聚集，使得查询检索过程更接近于人类信息检索的本质。文档重构的基本思想是根据每个用户查询的扩展词表，对文档的表达方式进行重新组织和替换，将文档中表示相同信息概念的单元聚集起来，统一用查询中的信息概念来表示，然后再进行检索。这个过程也是双向的，一方面将查询作为已知信息对文档进行重构；另一方面重构的范围则由查询扩展词表决定。

注意到在传统的查询扩展方法中，由于其本质是在扩展用户查询时把一个概念具体化，因此通常会把一个中心词的下义词也扩展进查询中。例如原始查询是“automobile”，则其下义词“sedan”“Ford”“Mazda”（其中 sedan 和 Ford 等处于不同的下义词级别中）等也都会被扩展进来形成新的查询。因此传统查询扩展中更关注下义词。而文档重构的思想可以看作是与查询扩展相逆的方法，其本质是把文档中含义相似的具体词集中到一个概念下。同样的例子中，使用文档重构方法，用户查询“automobile”不做改变，而在文档中的“sedan”、“Ford”、“Mazda”等词则替换为他们的上义词“automobile”。因此在文档重构的方法中，对于文档而言，我们关注的是词的上义关系是否能追溯到用户查询的中心词。

考虑到 WordNet 提供的词之间的语义联系是由人工构造的，其概念包含关系（上义、下义）和概念的不同表达方式之间的关系（同义）都比较可信，因此可以选择 WordNet 构造扩展词典。

文档重构是有确定目标的：如果一个文档中的词的同义词或者其 *n* 代上义词是某一个查询词，则把该文档中的相应词替换为查询词（即查询表达的信息概念）。一般只有名词才可能具有上义词组，因此，我们只需要对文档中的名词进行替换和重构，算法如下：

```
对文档 D 中的每一个名词 Wi{
    用 WordNet 扩展出同义词及 n 代上义词构成集合 Hype_n;
    对每一个查询集合 QuerySet 中的词 Kj{
        if（Kj 在 Wi 的 Hype_n 中）{
            在重构的文档 D'中写入 Kj;
        }else{
            在重构的文档 D'中写入 Wi;
        }
    }
}
```

算法 1 基于查询扩展的文档重构算法

随后，检索在文档 *D'* 上进行。

可以看到，在文档重构算法中，因为采用替换策略而非加权相加的策略，因此在重构过程中，对于非查询词来说，只有被替换或保留两种选择，不存在中间状态。同时被替换后的信息都是集中在查询词的统一概念下，因此也形成了对用户查询词的加强。这就将查询词和被替换词区别开来，也使得文档重构算法能够避免传统查询扩展中因为加权策略而引起的查询词和扩展词权值无法确定的问题。

文档重构方法较之查询扩展有两个优点：

1. 扩展有目的性，不存在从大量的被选词中确定扩展词的问题；

2. 用具有同义或者上义关系的查询词替换原词，相当于把扩展词和原词统一成一个概念，对查询词来说，文档扩展是在概念空间上进行索引和检索，不需要确定原始查询和扩展查询在计算相似度时的权重。

三．查询扩展与文档重构的比较分析

在前一节中对传统的查询扩展和本文提出的文档重构方法从信息的聚集和发散角度进行了描述。本节对两种方法进行进一步比较分析。从理论上来说，当使用同样的扩展词表，并且返回所有能够检索到的结果时，基于查询扩展与基于文档重构的检索能够找到的相关文档数是相同的。但是文档重构的方法能够改进相关结果文档的排序，使更有意义的相关文档排在更靠前的位置。

这里以向量空间模型中简单的相似度计算为例进行分析，其中文档权值计算使用传统的 *tfidf* 方法，查询项的权值直接用 *qtf* 表示。为简化计算，不妨设用户查询 Q 中只有一个查询词 T。设根据扩展词表，查询词项 T 有两个相似词 T_1 和 T_2 ，n、n_1 和 n_2 分别是文档数据集中出现了词项 T、T_1 和 T_2 的文档数，tf、tf_1 和 tf_2 分别为 T、T_1 和 T_2 在文档 D 中出现的词频。我们来考察一篇文档 D 与查询 Q 的相似度。（下文公式中的“ • ”表示向量内积，$\left\|\vec{X}\right\|$ 表示向量的模）

（1）如果不进行扩展，则有：

$$\vec{Q}=(w_{qT})=(qtf),\ \vec{D}=(w_{dT})=(tf\cdot idf)=\left(tf\cdot\log(\frac{N}{n}+1)\right),$$
$$sim(D,Q)=\vec{D}\bullet\vec{Q}=\frac{tf\cdot\log(\frac{N}{n}+1)\cdot qtf}{\left\|\vec{D}\right\|\cdot\left\|\vec{Q}\right\|} \tag{1}$$

（2）如果进行传统的查询扩展(Query Expansion, QE)，则有：

$$\vec{Q}'=(w_{qT},w_{qT_1},w_{qT_2})=(qtf,qtf_1,qtf_2),$$
$$\vec{D}=(w_{dT},w_{dT_1},w_{dT_2})=\left(tf\cdot\log(\frac{N}{n}+1),\ \lambda_1\cdot tf_1\cdot\log(\frac{N}{n_1}+1),\ \lambda_2\cdot tf_2\cdot\log(\frac{N}{n_2}+1)\right),$$
$$sim_{QE}(D,Q)=\vec{D}\bullet\vec{Q}'=\frac{tf\cdot\log(\frac{N}{n}+1)qtf+\lambda_1 tf_1\cdot\log(\frac{N}{n_1}+1)\cdot qtf_1+\lambda_2 tf_2\cdot\log(\frac{N}{n_2}+1)\cdot qtf_2}{\left\|\vec{D}\right\|\cdot\left\|\vec{Q}'\right\|} \tag{2}$$

设 S、S_1 和 S_2 分别为数据集中包含词项 T、T_1 和 T_2 的文档子集，并设 $n_{new}=|S'|=|S\cup S_1\cup S_2|$，（$|X|$为求集合 X 中的元素个数）其中 S’为经过文档重构后包含词项 T 的文档子集。

（3）如果进行基于查询扩展的文档重构(Document Refinement, DR)，则有：

$$\vec{Q}=(w_{qT})=(qtf),\ \vec{D}'=(w'_{dT})=(tf'\cdot idf)=\left((tf+tf_1+tf_2)\cdot\log(\frac{N}{n}+1)\right)$$
$$sim_{DR}(D,Q)=\vec{D}'\bullet\vec{Q}=\frac{(tf+tf_1+tf_2)\log(\frac{N}{n_{new}}+1)\cdot qtf}{\left\|\vec{D}'\right\|\cdot\left\|\vec{Q}\right\|} \tag{3}$$

考察查询扩展的相似度计算公式(2)。首先考察计算公式的分母，如果扩展出的新的查询词项太多，则新的查询向量的模$\left\|\vec{Q}'\right\|$会远大于原始查询向量的模$\left\|\vec{Q}\right\|$，加上扩展出的大量噪声，对相似度评

分造成极大的影响，因此在传统的查询扩展中，扩展的程度选择是一个重要而尚未很好解决的问题。

其次，考察公式（2）的分子。对于同一篇文档来说，tf_1 和 tf_2 与原始查询项的 *tf* 是基本可比的，因此主要的影响因素是新扩展出的词项的 *DF*（document frequency）值 n_1 和 n_2。因为在整个集合内对不同词项的使用情况是不确定的，因此很有可能出现新扩展词的 *DF* 值非常小的情况，于是扩展出的词项的 *tfidf* 权值远大于原始词项的权值。这样在计算相似度的时候，起决定性作用的将是新扩展出的 *DF* 值很小的那个词，而非含义相对更确定的原始查询项。这种情况将造成检索系统性能的不稳定。虽然新扩展词项的权值可以由 λ_1 和 λ_2 来调整，但是这种对每个扩展词都要确定不同 λ 值的方法在实际应用中是不可行的。这是传统的查询扩展方法中存在的第二个尚未很好解决的重要问题。

接下来分析文档重构方法中的相似度计算公式(3)。虽然经过重构后的文档向量的模 $\left\|\vec{D}'\right\|$ 会大于原始文档向量的模 $\left\|\vec{D}\right\|$，但是在一般情况下，因为文档向量的维数达到上千维，因此重构前后文档长度的差距不会很大，甚至可以忽略。因而最影响相似度评分的是公式（3）的分子部分。一般来说，经过重构后词项在文档中权值的 *tf* 因子增长的速率大于 *idf* 因子呈对数衰减的速率。因此在一般情况下，使用文档重构的方法会增强含有相关概念的文档的相似度评分，且性能较稳定。

因此，我们说根据语义关系查询扩展的文档重构方法，将文档中分散且表现为独立的、而实际上具有密切联系的概念聚集到了一起，符合真正信息和概念层次上的检索需要。

四．实时文档重构算法

基于语义关系查询扩展的文档重构方法虽然更接近人类检索信息的本质，但是算法 1 中描述的文档重构基本算法还无法直接应用于实时系统。这是因为根据这种基本思路，对于每个不同的用户查询，都要求对数据集合中的每篇文档进行一次重构和替换，其时间代价和复杂性均远远超过传统的查询扩展，因而在大规模的数据集合上是不可接受的。

但是分析发现，可以通过改变检索策略，通过使用一些近似计算，来实现文档重构的实时操作效果。实时文档重构的检索策略如下面的算法 2 所示。

(1) 对于查询关键字 *K*，利用 WordNet 扩展出同义词或者下义词集合{W_{K1}, W_{K2},……, W_{Kn}}；

(2) 不同于传统的扩展出新的查询，保持原始查询不变；

(3) 令 $(tf_K)_{new} = tf_K + \sum_{i=1}^{n} tf_{W_{Ki}}$

(4) 用$(tf_K)_{new}$代替不进行重构时的相似度计算公式中的 *tf*，仍然使用重构前的文档频度值 *n*，计算文档与查询之间的相似度；

(5) 对文档相似度进行排序，得到结果文档集合。

算法 2 基于查询扩展的实时文档重构检索策略

从算法 2 中可以看到，这种实时文档重构检索策略与文档重构基本思想（见公式 3）相比，只是在第四步中使用了一个近似：基本思想中，应该求得出现集合{*K*,W_{K1}, W_{K2},……, W_{Kn}}中任一词的所有文档集合 S_{new}，并计算新的文档频度值 $n_{new} = |S_{new}|$；但是在实时算法中，使用原始出现 *K* 的文档集合 *S* 中的文档数 $n = |S|$ 来近似 n_{new} 。这样整个实时算法与不进行重构相比，只需要多计算第三步中的$(tf_K)_{new}$ 的值，而这一计算是简单的求和，因此大大降低了文档重构时系统检索的复杂性，实现了实时的检索。

实时文档重构算法的关键是使用重构前出现查询词 *K* 的文档集合 *S* 中的文档数（即查询词 K 所

对应的原始 *DF* 值）来近似重构后出现查询词 *K* 的文档集合 S_{new} 的文档数（即查询词 K 所对应的新的 *DF* 值）。在信息检索模型中，一篇文档相对于用户查询的相似度，通常由两个因素决定：1.该文档与用户查询中的每个查询词之间的相关程度，通常与 *tf* 因素相关；2.用户查询中每个查询词本身的重要性，与 *DF* 相关。文档重构方法的目的是通过替换使表达同一个概念的文档与查询词之间的相关程度得到更多体现，即研究前一个因素对检索效果的影响。而查询词本身的重要性应该由它在整个文档集合中被实际使用的情况决定，是与具体的查询及其与文档的相似程度无关的信息，因此用重构前的原始 *DF* 信息更能真实反映用户查询中不同词项的重要性区别。因此用重构前的 *DF* 代替重构后的 *DF* 值是更合理的做法。

五．实验结果及分析

[实验组 1] 考察文档重构的有效性

在文本信息检索国际标准评测会议 TREC（Text REtrieval Conference）2002 年的 Novelty Track 标准测试集上进行实验。其官方评价标准为 F-度量值（F-Measure），辅助评价方法为 P×R（精度×召回率）。测试数据中的用户查询通常分为三个部分来描述：1. <title>部分，通常只有几个（一般不超过三个）词构成，是 Web 上使用的真实用户查询；2. <description>部分，对<title>部分的查询进行更详细的解释，说明用户想要寻找什么信息；3. <narrative>部分，对检索系统进行评价的标准给以说明，即找到什么类型内容的文档是相关的，什么样的是不相关的。一般来说在人们的研究中，根据这三个部分提取出三种类型的查询：短查询(short query，<title>部分)，中等查询(medium query，<title>+<description>内容)和长查询(long query，<title>+<description>+<narrative>内容)。

这里也分别使用了三种查询进行测试。文档重构追溯的上义词层数 *n* 分别取 0（即只考虑同义词，不考虑上义词），1（考虑同义词和一级上义词，以此类推），2，3，分别用 Synset，Hype_1，Hype_2，Hype_3 来表示。用来比较的基准是不进行扩展（Unexpanded）的检索结果。表 1 和表 2 分别是用 P×R 和 F 度量(F-measure)为评价标准的原始结果。表 3 和表 4 分别对应表 1、表 2，给出相对于不扩展提高的百分比。

表 1 基于查询扩展的文档重构检索结果(P×R)

	Unexpanded	Synset	Hype_1	Hype_2	Hype_3
短查询	0.057	0.063	0.065	0.064	0.064
中等查询	0.063	0.067	0.076	0.075	0.074
长查询	0.064	0.073	0.076	0.078	0.079

表 2 基于查询扩展的文档重构检索结果(F-measure)

	Unexpanded	Synset	Hype_1	Hype_2	Hype_3
短查询	0.172	0.180	0.184	0.181	0.182
中等查询	0.190	0.198	0.207	0.210	0.207
长查询	0.197	0.209	0.210	0.218	0.219

表 3 文档重构检索结果性能改进(P×R)

	Synset	Hype_1	Hype_2	Hype_3
短查询	+10.5%	+14.0%	+12.3%	+12.3%
中查询	+6.3%	+20.6%	+19.0%	+17.5%
长查询	+14.1%	+18.8%	+21.9%	+23.4%

表 4 文档重构检索结果性能改进(F-measure)

	Synset	Hype_1	Hype_2	Hype_3
短查询	+4.7%	+7.0%	+5.2%	+5.8%
中查询	+4.2%	+8.9%	+10.5%	+8.9%
长查询	+6.1%	+6.6%	+10.6%	+11.2%

可见无论对几级上义词进行考察，且无论使用哪种查询描述，基于查询扩展的文档重构在两种综合标准评价中都会大幅度提高检索性能。只用同义词集合进行重构的效果不如同时使用 *n* 级上义词的效果好。在扩展 3 级上义词并使用长查询的条件下，P×R 和 F-measure 都达到了最大值，分别比不扩展情况下得到的最佳结果提高了 23.4%和 11.2%，从而充分验证了文档重构的有效性。

[实验组 2] 文档重构与相应查询扩展的实验效果比较

在图 1 和图 2 中，我们给出文档重构与相应的传统查询扩展的比较结果。这里扩展词表均仍为 WordNet 中提供的同义和上/下义关系。其中 *DR*xx 和 *QE*xx 分别表示用文档重构(document refinement)和查询扩展(query expansion)的方法进行检索。

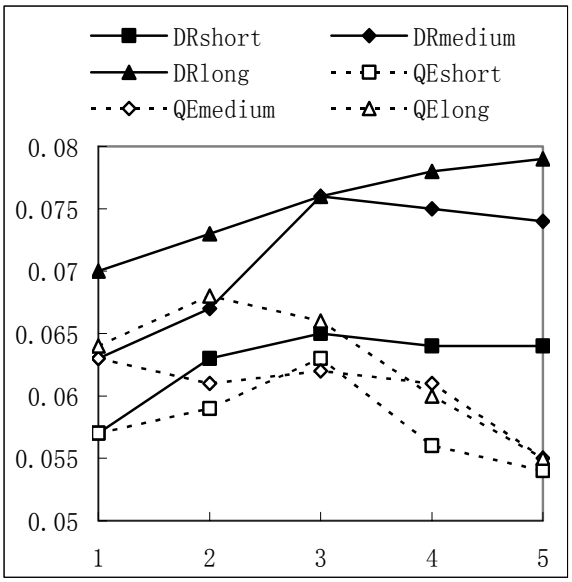

图 1 文档重构和查询扩展的比较（P×R）

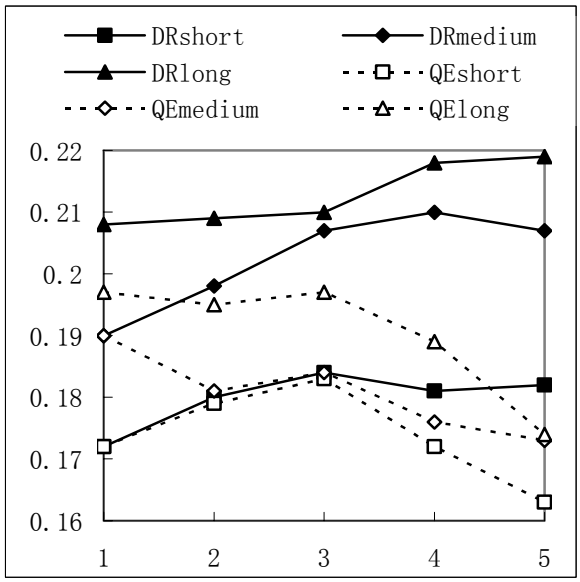

图 2 文档重构和查询扩展的比较（F-measure）

容易看到，用 WordNet 进行查询扩展相对于不扩展的提高并不稳定，在有些情况甚至比不扩展的结果要差，例如用中等查询描述来进行扩展和检索并使用 P×R 评价标准时得到的结果。而文档重构相对于查询扩展性能提高很明显。用 P×R 评价时，查询扩展的最佳结果是 0.068，而相应的（即使用同样的扩展词表）文档重构的最佳结果是 0.079，有相对 16.2%的提高；用 F-measure 评价时，查询扩展的最佳结果是 0.197，而相应文档重构的最佳结果是 0.219，有相对 11.2%的提高。

六．研究结论

为了解决信息检索中的词不匹配问题，本节提出了根据词之间的语义关系进行扩展和替换的文档重构方法。从理论上讲，它将扩展词和被扩展词合并成同一个概念进行检索，通过相关子信息的聚集，改进检索的效果。这种方法更接近于人类进行信息查找的思维过程。与传统的查询扩展不同，它不是把同一个信息概念下的不同词分散独立起来分别进行信息的匹配，而是将在文档中散布的表现为独立的、但实际上具有紧密语义概念联系的词聚集起来进行检索。分析表明，本文提出的文档重构方法，能够有效避免查询扩展方法带来的扩展层数和扩展词权值难以设定的两大问题。

进一步地，研究给出一种有效的文档重构的实时检索策略，从而解决了在实际应用中的检索可行性问题。

实验表明，以 P×R 为评价标准，基于查询扩展的文档重构方法不仅比不扩展的最佳性能有至少 14%到最大 23.4%的提高，而且比相对应的传统查询扩展方法也有大约 16%的提高；以 F-measure

为评价标准，文档重构方法比不扩展的最佳性能有至少 6%到最大约 11%的提高，而比对应的传统查询扩展方法的最佳性能有大约 11.2%的提高。尤其值得注意的是，使用文档重构的方法能够始终提高系统的总体检索性能，这不同于传统的查询扩展可能降低检索性能，同时体现了基于查询扩展的文档重构方法的可信性和有效性。

参考文献

[1] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to wordnet: An on-line lexical database. Technical report, http://www.cogsci.princeton.edu/wn/, 1993.

[2] Richardson R. and Smeaton A. Using WordNet in a Knowledge-Based Approach to Information Retrieval. Working paper CA-0395, School of Computer Applications, Trinity College Dublin. 1995.

[3] A.F. Smeaton and C. Berrut. Thresholding postings lists, query expansion by word-word distances and POS tagging of Spanish text. In Proceedings of the 4th Text Retrieval Conference, 1996.

[4] WordNet (a lexical database for the English language) homepage: http://www.cogsci.princeton.edu/~wn/

[5] HowNet knowledge hompage. http://www.keenage.com/ .

[6] Van Rijbergen, C. J. A theoretical basis for the use of co-occurrence data in information retrieval. Journal of Documentation (June 1977), pp 106-119.

[7] Crouch, C.J., Yong, B.. Experiments in automatic statistical thesaurus construction. SIGIR'92, 15th Int. ACM/SIGIR Conf on R&D in Information Retrieval, Copenhagen, Denmark, pp77-87, June 1992

[8] Schutze, H. and Pedersen, J.O. A cooccurrence-based thesaurus and two applications to information retrieval. In Proceedings of RIAO’94, pp. 266-274, 1994.

[9] H. Chen, B. Schatz, etc. Automatic thesaurus generation for an electronic community system. Journal of American Society for Information Science, Vol. 46, No. 3, pp. 175-193, 1995.

[10] Dekang Lin, Shaojun Zhao, Lijuan Qin and Ming Zhou. Identifying Synonyms among Distributionally Similar Words. To appear in Proceedings of IJCAI-03. 2003

[11] G. Ruge. Experiments on linguistically-based term associations. Information Processing and Management, Vol. 28, No. 3, pp 317-332, 1992.

[12] G. Grefenstette. Explorations in automatic thesaurus discovery. Kluwer Academic Publisher, 1994

[13] J. Xu and W.B. Croft. Query Expansion Using Local and Global Document Analysis. in Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 4-11, 1996.

[14] D. Lin. Dependency-Based Evaluation of MINIPAR. In Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation, Granada, Spain, May 1998.

[15] Lin D. and Pantel. P. Concept Discovery from Text. In Proceedings of Conference on Computa-tional Linguistics 2002. Taipei, Taiwan. pp577-583, 2002

[16] E. M. Voorhees. Query Expansion Using Lexical-Semantic Relations. In 17th Annual International ACM SIGIR conf., 1994