TREC 中提高检索鲁棒性的技术研究*

徐晋 赵军 徐波

中科院自动化所模式识别国家重点实验室 北京 100080

E-mail: {jxu, jzhao}@nlpr.ia.ac.cn, xubo@hitic.ia.ac.cn

摘　要：本文提出了两种提高检索鲁棒性的方法：(1) 词义熵权重计算公式；(2) 两级处理策略。在 NLPR-IR 信息检索系统上，以 TREC Robust 任务提供的大规模标准文本库（528155 篇文档，250 个公开的查询主题）为评测平台，检验了以上两个方法。实验表明，词义熵模型与当前常用的 TF*IDF 权重计算公式联合使用，能有效提高检索系统性能；而对两级处理策略，其也能有效地降低查询扩展中噪音对检索性能的影响。

关键词：信息检索，鲁棒性，词义熵，两级处理策略，TREC 评测

Study on Improvement of IR Robustness in TREC

Jin XU, Jun ZHAO, Bo XU

National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, 100080

E-mail: {jxu, jzhao}@nlpr.ia.ac.cn, xubo@hitic.ia.ac.cn

Abstract: This paper introduces two technologies to improve the Robustness of Information Retrieval: (1) word sense entropy; (2) two-step retrieval scheme. Upon our NLPR-IR system, based on the standard testing set (TREC Robust track, 528155 documents and 250 open user queries), the experiments have respectively compared the performances of the above two technologies. Our results show that, word sense entropy is efficient when it is combined with other weight measures such as TF, IDF; two-step retrieval scheme also proves helpful for reducing the influence of noise in query expansion.

Keywords: Information Retrieval, Robustness, Word Sense Entropy, Two-step Retrieval Scheme, TREC Evaluation.

1　TREC2004 Robust 任务介绍

TREC(Text Retrieval Conference)会议是国际上信息检索领域最重要的会议之一。它每年举办一次，主办者是 NIST (National Institute of Standards and Technology，美国国家技术标准局)和 DARPA(Defense Advanced Research Projects Agency，美国国防高级研究计划局)，其网址是：http://trec.nist.gov。TREC 评测是完全自愿，免费参加的，其目的在于，通过搜集大规模标准

*本文承国家自然科学基金（项目编号 60372016），国家自然科学基金（项目编号 60272041）和北京市自然科学基金（项目编号 4052027）的资助。

文本库和定义合理而全面的检索评测指标，来比较各个检索系统的性能。由于查询主题和文本库都是统一和公开的，并且不允许采用额外的训练语料，因此这个评测能够比较公正的反映出各个检索系统的算法性能。由于 TREC 评测的相对公正性，国内外许多大学和研究机构都通过参加自己感兴趣的任务，来测试自己的检索系统。包括：MIT, CMU, MSRA, IBM, 清华大学，复旦大学，中科院计算所等等知名的研究机构。

Robust 任务是 TREC 中一项子任务，它是 2003 年开始的，NLPR 参加了 2003 [2]以及 2004[5]两届的 Robust 任务。这个任务是传统文本检索任务的一个延续。检索的主要内容和形式都没有变化，还是根据查询在文本库中检索出相关的文本；但是，它主要是针对历年来在 TREC 中表现很差的查询主题（即所谓的 hard topic），从而重点评测参赛系统的鲁棒性；评测指标不再只是查准率，而开始关注系统的鲁棒性。下表所示 Robust 任务的评测指标[1]。

评测指标	具体描述
MAP	对所有的查询主题的平均准确率(Average Precision)的平均值
P@10	检索结果中，10 个相关文档被检索到时，所在位置以及排名更高的文档序列的检索准确率
TOP10	检索结果中，排名前 10 的文档没有一个是相关文档，符合这样条件的查询主题数目
AreaofC	最差的 x 个查询主题的 MAP(x)的曲线下的面积。这里，x 大小是从 1 到（0.25*查询主题总数）

对于 Robust 任务，我们根据其特点尤其是其对鲁棒性的要求，实验了如下两个技术：(1) 词义熵权重计算公式；(2) 两级处理策略。在接下来的章节中，第二章，对这两个方法的基本思想进行介绍；第三章，在 TREC 大规模标准文本库上实验和分析这两种方法；第四章，给出结论与未来工作的展望。

2 基本思想介绍

词义熵权重计算公式和两级处理策略这两种方法，前者主要是着眼于如何有效地、鲁棒地确定查询词的权重，而后者，则是着眼于如何充分利用查询词扩展的技术优势，同时又能有效地降低查询扩展带来的噪音对检索性能的影响。下面是两种方法的介绍。

2.1 词义熵权重计算公式

检索模型中，一个很重要的问题是如何确定每个查询词的权重（即查询词的重要性），常用的方法有 TF*IDF 权重公式[7]以及 BM25 算法[6]的权重公式。这些权重的公式一定程度地反映了查询项的重要性，在实际使用中也证明非常有效。但是，由于它们都是经验统计公式，所以严重依赖于特定的待检索文本库：对不同的检索文本库，某个词的权重可能会有很大不同，而且，由于文本库的不完整性（也就是文本库不能足够大到可以完整反映语言的统计特性），某些词的权重常常会很不可靠。这个问题实际上也是统计自然语言处理方法的一个通病。

举例来说，两个词’polio’和’bank’明显应该是有不同的权重，因为，直观看起来，’polio’对检索会有更好的指示性。但是，如果文本库中，这两个词正好都只出现了 1 次，那么他们的 DF 权重将会是一样的，都是 1。显然这个权重值不能真实的反映’polio’和’bank’这两个词的不同的重要性。为了克服这种由于统计数据的稀疏问题带来的不稳定性，我们自然想到，可不可以引入某种基于规则，或者是基于固定的词汇的语法语义的查询词计算公式，来提高权重计算的可靠性，从而提高检索的鲁棒性。

另一方面，我们也知道，选择每个词的最合适的词义对自然语言处理显然有很大的帮助，许多研究者把词义排歧引入到信息检索的工作中来。不过目前由于词义排歧的技术准确率还不高，而且，基于词义排歧的检索会加大系统的开销。所以，我们需要采用某些变通的方法来把词义选择的思想融入到检索系统中。

基于以上的分析，我们这里提出了词语词义多样性的权重计算公式—词义熵，它利用 Wordnet 中词义的信息特征对词的重要性进行打分。其基本思想如下所述：

Wordnet 是一个以语言心理学理论为基础的人工编制的词典参考系统[3]，Wordnet 对每个英文单词都有详细词义的解释，有些词中只有一个词义，而有些词在 Wordnet 中有几个词义。其中细节请参考 Wordnet 主页[3]。下面是一个具体的例子。

polio 的名词词义有 1 个

Overview of noun polio
The noun polio has 1 sense (first 1 from tagged texts)
1. (1) poliomyelitis, polio, infantile paralysis, acute anterior poliomyelitis -- (an acute viral disease marked by inflammation of nerve cells of the brain stem and spinal cord)

Bank 的名词词义有 9 个

Overview of noun bank
The noun bank has 10 senses (first 9 from tagged texts)
1. (883) depository financial institution, bank, banking concern, banking company -- (a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home")
2. (99) bank -- (sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents")
3. (76) bank -- (a supply or stock held in reserve for future use (especially in emergencies))
4. (54) bank, bank building -- (a building in which commercial banking is transacted; "the bank is on the corner of Nassau and Witherspoon")
5. (7) bank -- (an arrangement of similar objects in a row or in tiers; "he operated a bank of switches")
6. (6) savings bank, coin bank, money box, bank -- (a container (usually with a slot in the top) for keeping money at home; "the coin bank was empty")
7. (3) bank -- (a long ridge or pile; "a huge bank of earth")
8. (1) bank -- (the funds held by a gambling house or the dealer in some gambling games; "he tried to break the bank at Monte Carlo")
9. (1) bank, cant, camber -- (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)10. bank -- (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning); "the plane went into a steep bank")

表 1　　Wordnet 中的两个例子

可以看出，如果一个词有很多的词义，那么它的权重应该小一点，因为可能的情况是，比如，某篇文档含有这个词，但是它在这文档中的词义与查询中的词义不一样，那么即使包含了这个词，这篇文档也是与查询不相关的。所以，我们的权重计算原则应该如下。

如果一个词有较多的词义，那么它对检索贡献较小，相应权重也较小；
而如果一个词有较 少的词义，那么它对检索贡献较大，相应权重也较大。

根据这一原则，我们设计了词义熵权重计算公式：

$$H(word) = \sum_{i=1}^{n} p(sense_i \mid word) \log(p(sense_i \mid word))$$

$$p(sense_i \mid word) = \frac{c(sense_i, word)}{c(word)}$$

其中，$H(word)$ 是查询词 word 的词义熵；n 是查询词 word 在 Wordnet 中总计的词义数量；$sense_i$ 是查询词 word 在 Wordnet 中第 i 个词义；$c(sense_i, word)$ 是 word 的词义 $sense_i$ 在标记好的语料库中出现的次数；$c(word)$ 是查询词 word 在标记好的语料库中出现的总次数。

我们可以看出，基于 Wordnet 的结构化词义信息，词义熵在数学上完整地表达了词义多样性的原则。而词义熵的意义实际上和倒排文档频率(Inverted Document Frequency: IDF)的权重计算公式有相通之处，都是在表征词语对于检索的重要性。只是 IDF 是基于文本库的统计分析，而词义熵是基于 Wordnet 的人工标注的词义信息。在后面的实验中，我们也尝试了比较 IDF 与词义熵。

在实际的检索系统应用方案中，我们可以尝试用词义熵取代 IDF 权重计算公式或者把词义熵与别的权重如 TF、IDF 等联合在一起，用于计算文档的相似度。例如，对于最简单的 TF*IDF 矢量空间检索模型，可以有如下所示的两种方案来引入词义熵权重计算公式。

对于原始的TF*IDF，其相似度计算公式为 $R(q,d) = \sum_{word_j \in (q \wedge d)} tf(word_j) * idf(word_j)$。对于如何应用词义熵，第一个方案是替代IDF，单独使用词义熵权重计算公式，相似度计算公式为（以下简称为TF*Word Sense Entropy），$R(q,d) = \sum_{word_j \in (q \wedge d)} tf(word_j) * H(word_j)$；第二个方案是联合使用词义熵和IDF这两个权重计算公式，相似度计算公式为（以下简称TF*IDF*Word Sense Entropy）$R(q,d) = \sum_{word_j \in (q \wedge d)} tf(word_j) * idf(word_j) * H(word_j)$。这里，$R(q,d)$ 是查询 q 和文档 d 的相似度。在下节实验中，我们实验了以上这两种方案，并比较了这两种方案。

2.2 两级处理策略

我们知道，不管基于伪相关反馈还是基于语义知识，对查询词的扩展，一般都会引入许多与查询无关的查询词（我们称为噪音），这会很影响系统的性能[8]。所以，在 IR 研究中，研究者通常对查询词扩展持慎重的态度。我们提出两级处理策略希望能解决这个问题，提高检索系统的鲁棒性。如下是该策略的基本思路：

TREC 的 Robust 任务有如下两个重要特点：（1）每一个 TREC 风格的查询主题包含三个域的内容：标题(title)，描述(description)和详细叙述(narrative)。我们发现，标题域大都是名词，而且这些词有很强的指示性，对这个主题的检索很有帮助，我们称这些为“检索核心词”；而对于描述域和详细叙述域，他们常常是对标题域的进行更加细致的解释，有点类似于扩展词。（2）Robust 任务不需要对所有的相关文档都进行排序输出，而只需要输出 1000 篇文档。

根据这些特点，我们把检索过程分为两级：粗糙检索和精细检索，每级实现不同的目的，如下图所示。

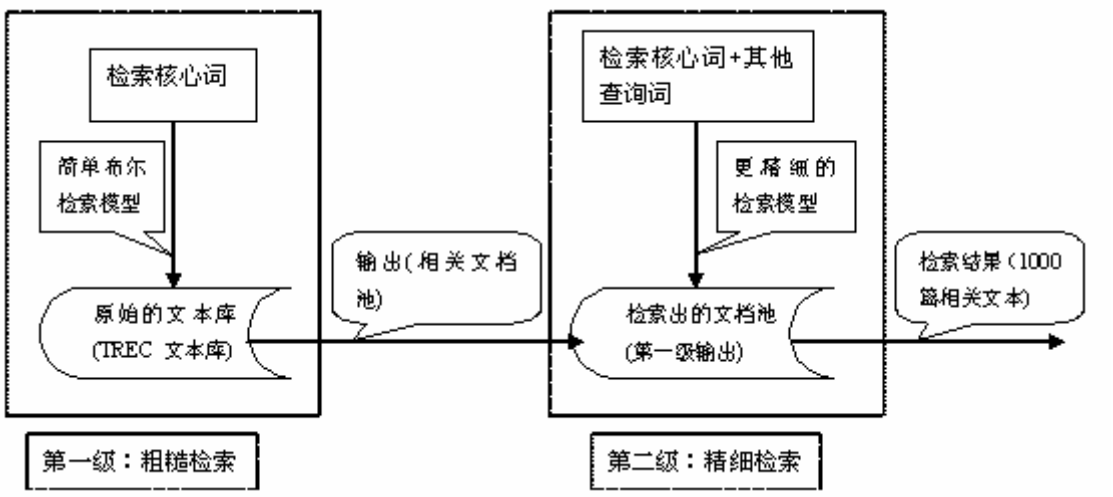

图 1　两级检索策略

Step 1　粗糙检索：以检索核心词作为输入，采用布尔简单检索模型进行检索。其中，在具体实施中，我们把查询主题的标题域中的词作为检索关键词，检索得到相关文档我们称为“相关文档池”，它将作为第二级检索的被检索文本库。这一级的目的是，尽可能把可能相关的文档都检索到，保证检索的召回率。

Step 2　精细检索：以检索核心词和其他的扩展词作为输入，以第一级得到的相关文档次作为被检索文本库，采用更为精细而复杂的检索模型进行检索，得到 1000 篇相关文本作为最后的检索结果。其中，在具体的实施中，我们把查询主题的三个域中的所有词、以及伪相关反馈得到的词，都作为查询词。这一级的目的是，在第一步的高召回率的基础上进行精细检索，输出 1000 篇相关文档，保证检索的排序准确率。

两级处理策略的好处是能提高检索的鲁棒性，有效地减少噪音的影响，因为第二级再使用精细检索方法，如相关反馈等查询扩展方法，虽然会有很多噪音，但是由于我们待检索的文本库是第一级的相关文档池，这些方法就不可能引入新的无关的检索文档，而只是影响相关文档池中各文档的排序。第二级方法可以理解为只是对第一级的相关文档池重新排序，并取出 1000 篇最相关的文档。

同时，我们可以看出两级处理策略另一好处：降低检索系统的开销。因为精细检索时，检索对象只是相关文档池，这里文档的数量远小于原始检索文本库中文档的数量。（比如，TREC　Robust 任务待检索的文本库有 5 0 多万篇文档，而通过第一级处理后，相关文档池一般有 2-3 万篇文档，两者差别大概是 25:1）。

3　实验结果比较和分析

实验中，我们采用了 TREC (http://trec.nist.gov)的 Robust Track 所使用的数据进行方法验证。其中数据文档集为 528155 篇文档，有 250 个用户查询主题(TREC topic 301-450 and 601-650 and 651-700)。需要注意的一点，由于 TREC2004 主办者的疏忽，在新出的查询主题 TREC topic651-700 中，有两个 topic 是重复的，所以实际上我们测试的是 249 个不同的查询主题。

3.1　词义熵实验

这里，我们使用了最常用的 TF*IDF 模型和我们自己提出的窗口模型作为实验的两个 baseline 系统(窗口模型是我们在 TREC2003 中提出的，细节请参考[4])，进行比较，以期检验词义熵是否能同时改善两个模型的性能。其中，我们实验了用词义熵来替代 IDF 权重公式，以及联合使用词义熵和 IDF 两个权重计算公式的两套方案，实验结果如下。

ID tag	MAP	P(10)	Top10	AreaofC
TF*IDF	0.1617	0.2546	31	0.0088
TF*WE	0.1458	0.2530	29	0.0081
TF*IDF*WE	0.1720	0.2594	29	0.0094
Windows	0.2043	0.2958	25	0.0103
Windows*WE	0.1921	0.2802	24	0.0112
Windows*IDF*WE	0.2179	0.3136	24	0.0117

表2　　词义熵的实验结果比较

上表中标号的解释：（1）TF*IDF 是 TF*IDF 模型的检索结果；（2）TF*WE 是替代 IDF，单独使用词义熵模型的 TF*IDF 模型的结果；（3）TF*IDF*WE 是联合使用词义熵和 IDF 的 TF*IDF 模型的结果；（4）Windows 是窗口模型的结果；（5） Windows*WE 是替代 IDF，单独使用词义熵模型的窗口模型的结果；（6）Windows*IDF*WE 是联合使用词义熵和 IDF 的窗口模型的结果。

从表 3 可以看到，（1）用词义熵完全取代 IDF 后，实验效果并没有比原来使用 IDF 好，其中可能的一个原因是，我们现在的词义熵公式调整得还不够好，可能与 TF 等计算公式没有归一到一个水平，这样会影响权重结果，我们知道，IDF 实际上也是几经演化，并调整了参数后才形成目前大家公认的计算方法。（2）联合使用词义熵和 IDF 权重，实验结果比 baseline 系统好，而且是对 TF*IDF 模型和窗口模型这两个模型都有提高，这说明对于词义熵的计算模型，虽然还不成熟，但其指导思想还是可行的，实验证明它是可以提高检索模型的性能，增强检索系统的鲁棒性。

3.2　两级处理策略实验

这里，我们使用了上一节中表现最好的窗口模型(Windows*IDF*WE)作为实验的 baseline 系统，分别实验了两级处理策率对于伪相关反馈和简单扩展三个域的作用效果。实验结果如下。

ID tag	MAP*	P(10)*	Top10*	AreaofC*
Windows	0.2179	0.3136	24	0.0117
Windows+all fields	0.1826	0.3025	22	0.0125
Windows+pseudo feedback	0.2263	0.3398	25	0.0114
Windows+all fields +two step	0.2215	0.3601	24	0.0136
Windows+all fields +pseudo feedback+two step	0.2438	0.4137	22	0.0141

表3　　两级处理策略的实验结果比较

上表中标号的解释：（1）Windows 是上一节中联合使用词义熵和 IDF 的窗口模型的结果（Windows*IDF*WE）。（2）Windows+all fields 是使用了全部三个域(title, description, narrative)

的结果；（3）Windows+pseudo feedback 是使用了伪相关反馈进行查询扩展的结果；（4）Windows+all fields +two step 是采用两级处理策略后 Windows+all fields 的结果；（5）Windows+all fields +pseudo feedback+two step 是采用两级处理策略后 Windows+all fields+pseudo feedback 的结果。

从表 4 可以看到，（1）采用伪相关反馈或者采用全部三个域，都不能很明显地提高系统的性能，甚至使用全部三个域，系统性能还有下降。（2）使用两级处理策略后，对于两种扩展方法，都有明显的提高，这有力地证明了采用两级处理策略对于降低噪音影响的积极意义。

4 结论

为了提高检索鲁棒性，本文提出了两种新方法：(1) 词义熵权重计算公式；(2) 两级处理策略。在 TREC Robust 任务提供的大规模标准文本库（528155 篇文档，250 个公开的查询主题）的评测平台下，我们检验了以上两个方法。实验表明，词义熵模型与常用的 TF*IDF 权重计算公式联合使用时，能有效提高检索系统性能；而对两级处理策略，其也能有效地降低查询扩展中噪音对检索的影响。

未来的工作：（1）从实验中，我们可以看到，词义熵并不能完全取代 IDF 的作用，分析原因，可能是因为我们现在的词义熵公式调整得还不够好，还需要进一步优化。这需要我们做更多的公式比较实验，从而得出一个比较优化的词义熵计算公式。（2）而对于两级处理策略，我们也需要在更多的比较实验中证明其有效性。需要指出的是，实验中，我们采用伪相关反馈并没有得到某些文献中所提到的大幅性能提升，这可能是因为我们的反馈公式中的参数选择不够好，为有力的说明两级处理策略的有效性，我们还需要更细致的实验。

参 考 文 献

[1] http://trec.nist.gov

[2] Qianli Jin, Jun Zhao, Bo Xu. NLPR at TREC 2003 – Novelty and Robust Track. Text Retrieval Conference (TREC-12), NIST, Maryland, USA, 2003.

[3] http://www.cogsci.princeton.edu/~wn/

[4] Qianli Jin, Jun Zhao, Bo Xu, Window-based Method for Information Retrieval, 2004, The First International Joint Conference on Natural Language Processing

[5] Jin Xu, Jun zhao, Bo Xu, NLPR at TREC 2004: Robust Experiments, Text Retrieval Conference (TREC-13).

[6] S.E. Robertson, S.Walker. Okapi/Keenbow at TREC-8. Text Retrieval Conference, NIST Specail Publication 500-246, (1999).

[7] ROBERTSON, S.E. and SPARCK JONES, K., 'Relevance weighting of search terms', Journal of the American Society for Information Science, 27, 129-146 (1976).

[8] Robertson, S.E. On term selection for query expansion. Journal of Documentation 46, Dec 1990, p359-364.