확률적 정보 검색 모델에서의 유사 적합성 피드백 실험

조봉현° 이창기 안주희 이근배
포 항 공 과 대 학 교 컴 퓨 터 공 학 과
{bhcho, leeck, ahnjh, gblee}@nlp.postech.ac.kr

Experiments on Pseudo Relevance Feedback in Probabilistic Information Retrieval Model

Bong-Hyun Cho° Chang-kee Lee Joo-Hui An Gary GeunBae Lee
Dept. of Computer Science & Engineering, POSTECH

요 약

본 논문은 확률기반 자연어 검색 시스템 POSNIR/E를 이용한 여러 가지 유사 적합성 피드백 방법들이 검색 시스템의 성능 향상에 기여할 수 있는 정도를 보여주고, 확률 기반 정보 검색 시스템에 적합한 유사 적합성 피드백 수행 방법을 제시한다. POSNIR/E는 한국어 자연어 검색 시스템, POSNIR를 기반으로 만들어진 영어 자연어 검색 시스템이다. 이 시스템은 성능 향상을 위한 질의 확장의 방법으로 검색 단계에서 유사 적합성 피드백을 적용한다. 검색 단계에서 영어 태거에 의해 태깅된 사용자 질의로부터 질의어를 추출하고 초기 검색을 수행한다. 유사 적합성 피드백을 위하여 초기 검색 결과 중 상위 5개의 문서에 나타나는 키워드를 중요도에 따라 내림차순 정렬하여 상위 10개의 키워드를 초기 질의어에 확장한다. 이렇게 확장된 질의어로 최종 검색을 수행한다. TREC 평가용 테스트 컬렉션 WT10g와 TREC-9의 질의-적합문서 집합을 이용하여 여러 가지 TSV 함수를 사용하여 검색 성능을 평가 하였다. 실험 결과 유사 적합성 피드백을 사용할 경우 TSV 함수에 확률 모델의 CF 요소 뿐만 아니라 TF 요소 등을 적용 시킬 경우 성능 향상에 기여할 수 있음을 알 수 있었다. 또한 색인어와 검색어로 단일어 뿐만 아니라 복합어도 사용할 경우 성능이 향상됨을 알 수 있다.

1. 서론

이상적인 정보검색 시스템은 사용자 질의의 내용을 완벽하게 이해하여 문서 집합으로부터 사용자 요구에 가장 부합할 수 있는 문서를 검색해 주는 것이다. 그러나 입력된 질의로부터 사용자 의도를 완벽하게 이해하기는 쉽지 않다. 그 이유는, 실제 질의는 사용자 의도를 충분히 예측할 수 있을 만큼 길이가 길지 않거나(보통 3개 미만의 워드), 질의로부터 추출된 질의어의 단순 매칭에 의한 문서 검색은 항상 사용자 요구를 만족시키기에 충분하지 않기 때문이다. 때문에 여러 연구자들에 의해 사용자 질의를 확장 할 수 있는 방법들이 연구되어 왔다. 그 중에서 적합성 피드백 수행이 성능을 향상시킴이 여러 연구에서 밝혀졌다[1][3][8]. 적합성 피드백 방법은 초기 검색 결과에 대해 사용자로부터 적합성 여부에 대해 피드백을 받아 초기 질의를 수정하여 재 검색을 하는 과정을 통해 검색 성능을 향상시킨다. 즉 사용자 피드백이 없다면 초기 질의에 대한 수정을 할 수 없다는 단점이 있다.

이러한 문제점을 해결하기 위하여 제시된 방법이 유사 적합성 피드백이다. 유사 적합성 피드백은 사용자의 피드백 없이 사용자 질의에 대한 초기 검색 결과로부터 적합성 정보를 얻어 초기 질의를 변형한 후 재 검색하여 검색 성능을 향상시키는 방법이다.

*본 연구는 교육부 BK21 Project 지원으로 수행되었음.

유사 적합성 피드백을 통한 질의 변형 방법에는 크게 두 가지의 논점이 있다. 첫번째는 질의 확장과 질의어에 대한 재 가중치 부여에 관한 것이고, 두 번째는 어떻게 질의 확장 대상 키워드를 선별할 것인가에 관한 것이다.

첫번째 논점은, 질의 변형의 방법으로 질의어에 대한 가중치만 다시 부여할 것인가, 또는 질의어 가중치를 다시 부여를 하지 않고 새로운 키워드로 질의 확장만 할 것인가, 또는 두 가지를 동시에 할 것인가에 관한 것이다. 벡터 모델의 경우, Rocchio식[1]과 같이 질의 확장과 재가중치 부여를 동시에 수행할 수 있는 방법으로 체계적인 방법이 제시되었지만, 확률 모델에 관해서는 아직까지 그렇지 못하다.

두 번째 논점은, 어떤 키워드를 사용하여 질의 확장을 할 것인가에 관한 것이다. 벡터 모델의 경우 기본적으로 검색된 적합 문서에 존재하는 모든 키워드를 확장 대상어로 간주한다. 그러나 적합 문서에 존재하는 일부 선별적인 키워드로 주어진 질의를 확장하는 것이 모든 키워드로 확장하는 것보다 더 좋은 성능이 있음이 Harman[8]에 의해 밝혀졌다. 확장되는 키워드의 개수는 검색 시간에도 중대한 영향을 미치기 때문에 적절한 개수의 키워드 선별은 매우 중요하다.

본 연구에서는 확률 모델에서 유사 적합성 피드백의 여러 가지 방법을 제시하고 실험하였다. 실험 결과, 유사 적합성 피드백을 위한 TSV(Term Selection Value) 함수에 기존에 많이 사용 되어 오던 ICF(Inverted Collection Frequency) 요소 뿐만 아니라, TF(Term Frequency) 요소 및 문서 길이 정보 등 부가적 정보를 적용시켰을 때 성능이 향상됨을 알 수 있었다. 또한 단일어 뿐만 아니라 복합어(noun phrasal terms)를 색인어와 질의어로 사용했을 경우 성능이 향상됨을 확인할 수 있다.

2장에서는 유사 적합성 피드백에 대한 관련 연구 내용을 기술 했으며, 3장에서 확률 모델에 대한 소개와 색인 과정, 검색 과정 등 POSNIR/E 시스템 전반에 관한 내용을 다룬다. 특히 3장에서는 여러 가지 TSV 함수 등에 대해 자세히 소개한다. 4장에서는 TREC 평가용 문서 집합인 WT10g와 TREC-9의 질의-적합 문서 집합을 이용한 실험 결과 및 분석에 대해 기술한다. 마지막으로 5장에서 본 연구의 결론과 향후 계획을 제시한다.

2. 관련 연구

2.1 벡터 공간 모델(The Vector Space Model)

원래 적합성 피드백 처리는 1960년 초반 Rocchio에 의해 벡터 모델에 적용 되기 시작하였다[1]. 벡터 공간 모델이란, 문서 D와 질의 Q를 n-차원의 벡터로 표현하고 그 내적의 합을 구하여 문서-질의간 유사도를 측정하는 모델이다.

$$D = (d_1, d_2, \cdots, d_n)$$

$$Q = (q_1, q_2, \cdots, q_n)$$

$$Sim(D,Q) = \sum_{i=1}^{t} d_i \bullet q_i$$

이후 Ide는 Rocchio식을 발전시켰고[2], 1990년에 Salton & Buckley에 의해 제안된 세가지 식에 대해 비교 실험이 이루어 졌다[9]. 6개의 실험 문서 집합에 대한 이 실험에서, Ide dec-hi 방식이 전체적으로 가장 좋은 성능을 보였으나 다른 방식과의 성능 차이가 크게 나지 않았다. Rocchio와 Ide 에 의해 제시된 방법은 기본적으로, 문서 벡터와 질의 벡터를 합치면서 자동으로 질의 가중치 재 조정과 확장을 동시에 할 수 있다.

Mitra는 적합성 피드백 수행에서 사용되는 적합 문서로 가정되어 사용되는 문서의 정제가 성능 향상에 기여함을 보였다[12]. 이것은 적합 문서라고 가정된 문서에 포함된 부적합 문서로부터 추출된 확장어가 성능을 저하시키기 때문이다.

2.2 확률 모델(The Probabilistic Model)

확률 모델은 적합 문서와 부적합 문서에 나타나는 질의어의 분포에 기초해 Robertson & Sparck Jones에 의해 제안된 모델이다[3].

$$w^{(1)} = \log \frac{p(1-q)}{q(1-p)}$$

Sparck Jones는 이러한 상대적 가중치 식을 이용하여 적합성 피드백을 수행에 적용하였다[5]. 이 실험에는 사용자들로부터 적합 문서로 피드백 받은 문서를 이용하여 초기 질의어에 대한 가중치를 다시 부여를 했는데, 일반적 IDF값을 이

용할 경우보다 높은 성능을 보여, 적합성 피드백을 위해서는 확률적 가중치 방법이 유용함을 보여 주었다.

Croft & Harper는 사용자 피드백을 받는 대신 초기 검색에 의해 검색된 문서를 이용하여 초기 질의어의 가중치를 다시 부여하는 방법을 제시했다[6].

확률적 가중치 방법 자체는 질의를 확장할 수 있는 스킴을 제공하지 않는다. 그러나 많은 연구자들은 질의 확장에 확률적 가중치 방법을 사용하려고 여러 가지 시도를 했다.

Harper & van Rijsbergend는 초기 검색의 적합 문서를 이용해 질의어 가중치를 재조정하고, MST(Maximum Spanning Tree)를 이용하여 질의어에 직접 연결된 모든 키워드를 초기 질의에 확장해 주는 방법을 제시하였다[4].

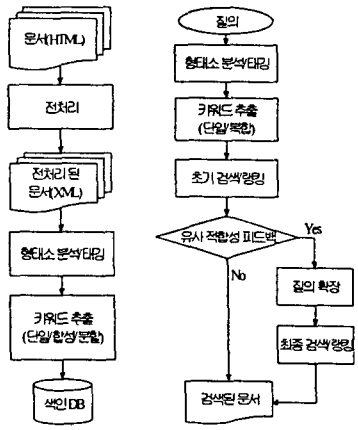
Wu & Salton은 Cranfield collection를 사용한 실험에서, 초기 검색에 의한 적합 문서를 이용한 질의어 재가중치 부여만으로 27%의 정확도 향상과, 재가중치 부여와 함께 적합 문서의 모든 키워드를 확장어로 사용하여 질의를 확장했을 때 32.7%의 정확도 향상이 있음을 보였다[7].

3. 시스템 개관

POSNIR/E는 한국어 자연어 정보 검색 시스템 POSNIR[13]를 기반으로 영어 자연어 정보 검색을 위해 만들어진 확률 기반 정보 검색 시스템이다. 본 시스템은 크게 색인 모듈과 검색 모듈로 구분할 수 있다[그림 1].

3.1 색인 모듈

색인 모듈에서는 입력된 문서를 전처리 과정을 통해 의미 없는 필드와 HTML Tag 등을 제거 한 후 XML형태의 문서로 변환한다. 변환된 XML 형태의 문서를 품사 태거로 문서의 각 문장을 태깅한 후 명사, 형용사, 동사를 키워드로 추출한다. 또한 이 과정에서 품사 태그 패턴에 기반한 복합어를 생성하여 색인한다.



[그림 1] POSNIR/E 시스템의 색인 및 검색 모듈

3.1.1 형태소 분석 / 품사 태거

본 시스템은 포항공대 자연어 처리 연구실에서 개발한 품사 태거를 사용했다. 포항공대 자연어 처리 연구실의 POSTAG/E는 upenn 품사 집합을 사용하고 HMM 기반의 품사 태거이다. POSTAG/E는 크게 2개의 구성 요소로 이루어져 있다. 첫번째 전처리 부분, 이 부분에서는 문자열을 입력으로 받아 문장을 인식하고 영어의 Word를 태깅 단위로 분리한다. 두 번째는 형태소 분석/태깅 단계, 이 단계에서는 사전에서 태깅 단위를 찾고 최적의 품사를 결정한다. 태거의 출력은 입

입력 문장 : There is no asbestos in our products now.

형태소 분석/태깅 결과 :

```
There      BOS   =₩₩EX
is                 be₩VBZ
no                 =₩DT
asbestos           =₩NN
in                 =₩IN
our                =₩PRP$
products           product₩NNS
now                =₩RB
.          EOS     =₩.
```

력 문자열, 품사, 원형으로 이루어져 있다. 따라서 이 결과를 받아서 원하는 정보를 사용하면 된다. 다음은 실행 결과이다.

3.1.2 2단계 색인어 추출 (2-phase index term extraction)

색인어 추출은 영어 형태소 분석 및 품사 태거에 의해 출력된 키워드들 중에서 명사, 형용사, 동사 품사를 갖는 키워드와 인용구 등으로 둘러 쌓인 키워드(영화 제목, 책 제목 등)를 대상으로 한다.

태깅의 결과인, 입력문자열, 품사, 원형 중에서 원형(lemma)을 색인 대상으로 했으며, 원형이 사전에 등록 되어 있지 않은 키워드에 대해서는 색인어-질의어 간의 불일치를 해결하기 위해 Stemmer[14]를 사용하여 stemming결과를 색인 대상으로 하는 2단계 색인어 추출 방법을 사용하였다.

색인어로 추출된 각 키워드는 단일어(Single terms)로서 색인 DB에 저장한다. 이 과정에서 품사 태그 패턴에 기반하여 생성된 복합어(Noun Phrasal Terms)를 색인 DB에 같이 저장하는데, 복합어 후보로 가능한 품사 태그 패턴으로는 다음과 같은 패턴을 사용하였다. 그리고, 복합어를 구성하는 단일어의 개수는 3개 이하로 제한하였다.

2개의 단일어로 구성된 복합어

Term1/{NN | NP} Term2/{NN | NP}

→ Term1_Term2

Term1/{NN | NP} (' s/POS | of/IN) Term2/{NN | NP}

→ Term1_Term2

3개의 단일어로 구성된 복합어

Term1/JJ Term2/{NN | NP} Term3/{NN | NP}

→ Term1_Term2_Term3

Term1/JJ Term2/JJ Term3/{NN | NP}

→ Term1_Term2_Term3

3.1.3 불용어 제거

추출된 단일어 중에 불용어 리스트에 속한 키워드는 색인어에서 제외된다. 복합어의 경우, 복합어를 구성하는 단일어 중에서 제일 마지막 키워드가 불용어일 경우 복합어를 생성하지 않는다.

3.1.4 색인 DB 압축 및 분할

색인 과정에서 각 색인어의 TF, DF를 구하기 위하여 색인 DB의 검색, 삽입, 수정이 반복적으로 나타나게 된다.

본 시스템에서는 B+ Tree구조의 색인 DB를 사용하는데, 색인 과정에서의 DB 연산에 의해 색인어의 개수가 늘어 날수록 DB 접근 시간(DB Access Time)이 지수배로 증가하는 현상이 발생하였다. 따라서 문서 집합을 적당한 크기로 나눈 후 나뉘어진 각 문서 집합을 색인하는 분할 색인 방법을 사용하였다. 또한 각 분할 색인 DB에 반복적인 자연수로 저장되는 정보(DF, TF, FC, TT)를 색인 모듈에서 인코딩하고 검색 과정에서 디코딩하여 이용함으로서 색인 DB의 크기를 줄이는 방법을 사용하였다. 여기서 FC는 Filed Constraint로 색인어가 추출되는 문서내의 위치(또는 필드) 정보이고, TT는 Term Type으로 추출된 색인어가 단일어인지 복합어인지에 관한 정보이다.

3.2 검색 모듈

3.2.1 질의어 추출

질의 추출 단계에서는, 자연어 형태의 사용자 질의에서 형태소 분석, 태깅을 통해 명사, 형용사, 동사와 인용구 등으로 둘러 쌓인 키워드를 질의어로 추출한다. 또한 이 과정 중에 색인 모듈의 색인어 추출과 마찬가지로 복합어를 생성하여 질의어 리스트에 추가한다.

3.2.2 질의 대상 불용어 제거

추출된 질의어 리스트에서 색인어 추출에서와 마찬가지로 불용어를 제거한다. 또한, "find", "document", "identify" 등과 같이 질의 문장에 특수한 키워드를 추가 제거한다. 이러한 질의 대상 불용어는 실제 질의 로그에서 얻은 30여 개의 키워드 리스트이다.

3.2.3 2-포아송 모델(2-Poisson Model)

본 시스템은 확률 분포에 기반한 Robertson의 2-포아송 모델을 사용한다. 가장 기본적인 확률 모델은 Robertson/Spark Jones에 의해 제안된 식(1)과 같다. 식(1)에서는 질의어의 문서의 출현 여부만을 고려하였는데, 본 시스템에서는, 문서의 출현 빈도(Term Frequency), 문서 길이(Document Length) 등을 고려한 식(2)와 같은 Okapi BM25 함수[10]를 사용하였다. 검색 모듈에서는 주어진 질의에 대

해 식(2)에 의해 각 문서의 Score를 구한다.

$$w^{(1)} = \log \frac{p(1-q)}{q(1-p)} = \log \frac{(r+0.5)(S-s+0.5)}{(R-r+0.5)(s+0.5)} \quad (1)$$

$$Score(d,q) = \sum_{\substack{term\_t \\ in\_q}} \left(\frac{(k_1+1)\times tf_t}{k_1 \times ((1-b)+b\times \frac{dl_d}{avdl}) + tf_t} \times \log \frac{N-df_t+0.5}{df_t+0.5} \times \frac{(k_3+1)tf_q(q,t)}{k_3+tf_q(q,t)} \right) \quad (2)$$

p: 질의어가 적합 문서에 나타날 확률

q: 질의어가 부적합 문서에 나타날 확률

N: 전체 문서 수

n: N 중에서 질의어를 포함하는 문서 수(=df_t)

R: 질의 Q에 대해 적합하다고 판단되는 문서 수

r: R 중에서 질의어 t를 포함하는 문서 수

S: 질의 Q에 대해 부적합 하다고 판단되는 문서 수

s: S 중에서 질의어 t를 포함하는 문서 수

tf_t: 질의어 t의 한 문서 내 출현 빈도

k_1, b, k_3: 상수 파라메터

dl_d: 문서 d의 길이

$avdl$: 전체 문서의 평균 길이

식(2)는 3가지 구성 요소, 즉, 첫번째의 TF(Term Frequency) 요소, 두 번째의 ICF(Inverted Collection Frequency)요소, 세 번째의 QTF(Query Term Frequency)요소로 구별될 수 있다.

3.2.4 유사 적합성 피드백

검색 성능을 높이기 위한 질의어 확장과 재가중치 방법으로 널리 사용 되고 있는 방법이 유사 적합성 피드백이다. 유사 적합성 피드백 적용의 적용 절차는 다음과 같다.

질의어 추출 단계

입력 받은 사용자 질의로부터 질의어를 추출하여 질의어 리스트를 만든다. 질의어로는 단일어 뿐만 아니라 복합어 또한 추출된다.

초기 검색 단계

가중치가 부여된 질의어 리스트의 각 질의어로 식(2)에 의해 해당 문서를 검색하고, 검색된 각 문서를 Score에 따라 랭킹한다.

질의 확장 단계

랭킹된 문서 집합으로부터 상위 R개의 문서를 적합 문서로 가정하고 그 문서에 존재하는 모든 키워드에 대해 적절한 함수를 사용하여 TSV(Term Selection Value)를 구하여 그 값에 따라 랭킹한다. TSV가 높은 상위 K개의 키워드를 원 질의어 리스트에 추가한다.

Harman은 여러 가지 TSV 함수를 사용하여 유사 적합성 피드백을 적용했는데, Robertson과 Sparck Jones에 의해 제시된, 질의어가 나타나는 적합 문서와 부적합 문서의 확률 분포로부터 추정된 식(1)을 사용하였을 때 가장 높은 성능을 보였다[8].

하지만 식(1)은 주어진 질의에 대한 모든 적합 문서 정보를 미리 알고 있음을 가정하였다. 그러나 일반적으로는 주어진 질의에 대하여 일부분의 적합 문서 정보만을 알 수 있다. 이러한 이유로 본 논문에서는 Robertson과 Walker에 의해 제시된 식(3)를 TSV 함수로 사용하였다[11].

$$w^{(1)} = TSV = \frac{k_5}{k_5+\sqrt{R}}\left(k_4 + \log\frac{N}{N-n}\right) + \frac{\sqrt{R}}{k_5+\sqrt{R}}\left(\log\frac{r+0.5}{R-r+0.5}\right) - \left(\frac{k_6}{k_6+\sqrt{S}}\log\frac{n}{N-n}\right) - \left(\frac{\sqrt{S}}{k_6+\sqrt{S}}\log\frac{s+0.5}{S-s+0.5}\right) \quad (3)$$

N, n, R, r, S, s는 식(2)에서와 동일하고, k_5, k_6은 각각 상수 파라메터이다. 위 식은 적합문서 정보가 없을 경우의 w_p에 대한 추정치(log(N/(N-n))와, 적합문서 정보가 있을 경우의 w_p에 대한 추정치(log((r+0.5)/(R-r+0.5))의 조합이고, k_5에 의해 두 추정치의 가중치가 조절된다. w_q에 대한 두 추정치(각각 log(n/(N-n)), log((s+0.5)/S-s+0.5)))에 대해서도 w_p와 마찬가지로 조합하여 k_6로 가중치를 조절한다. 그런데, 식(3)을 TSV 함수로 그냥 사용했을 경우, 확장 대상 키워드의 tf값이 너무 낮아지는 문제가 발생했고, 본 연구에서는 이러한 문제를 해결하기 위하여 tf를 반영한 식(4),(5),(6)을 TSV 함수로 사용하고, 각각 비교 실험을 하였다.

$$TSV = \left(\sum_{d \in R} 0.5 + 0.5 \times \frac{tf_{t,d}}{dl_d} \right) \times w^{(1)} \quad (4)$$

$$TSV = \left(\sum_{d \in R} \log(tf_{t,d} \times \frac{avgdl}{dl_d} \right) \times w^{(1)} \quad (5)$$

$$TSV = \left(\sum_{d \in R} \frac{tf_{t,d}}{k_1((1-b)+b\frac{dl_d}{avgdl})+tf_{t,d}} \right) \times w^{(1)} \quad (6)$$

$w^{(1)}$은 식(3)의 $w^{(1)}$이다. 위 식은 문서 길이로 정규화 된 tf를 여러 가지 함수 형태로 반영한 것이다. 특히 식(6)은 2-포아송 모델의 Score 식에서, TF 요소와 적합 문서 정보를 반영한 ICF요소를 사용한 것이다.

최종 검색 단계

확장된 질의어 리스트를 이용하여 최종 검색을 한다. 이 단계에서는 초기 검색 모델식과는 달리, 적합문서 정보를 이용할 수 있다. 따라서 식(7)을 사용하여 질의어가 나타나는 각 문서에 Score를 부여한다.

$$Score(d,q) = \sum_{\substack{term\_t \\ in\_q}} \left(\frac{(k_1+1)\times tf_t}{k_1\times((1-b)+b\times\frac{dl_d}{avdl})+tf_t} \right. \\ \left. w^{(1)} \times \frac{(k_3+1)tf_q(q,t)}{k_3+tf_q(q,t)} \right) \quad (7)$$

$w^{(1)}$은 식(3)의 $w^{(1)}$이며, 나머지는 식(2)와 동일하다.

4. 실험 및 분석

앞 섹션에서 제시한, 단일어 유사 적합성 피드백을 위한 여러 가지 TSV 함수의 성능 평가를 위하여 TREC 데이터를 사용하였다.

4.1 문서 집합 색인

문서 집합으로 TREC 데이터, WT10g를 사용 하였다. WT10g는 실제 웹 검색 환경에서의 성능 평가를 위해 만들어진, 전체 5,157개 파일, 1,692,096개의 문서로 이루어진 10GB 크기의 문서 집합이다.

전처리 과정을 통해 색인에 필요 없는 HTML Tag 등을 제거하여 약 6.5GB 크기의 XML형식의 문서 집합을 대상으로 색인하였다. 색인 과정에서 전체 문서 집합을 약 200MB 단위로 나누어 총 26개로 분할 색인하였다. 각 문서에서 <TITEL>, <BODYTEXT> 필드를 대상으로 색인어를 추출하였고 색인어가 추출된 필드에 따라 색인어의 가중치를 다르게 주었다.

모든 문서를 색인 하는데 대략 10일이 소요되었으며, 색인 DB는 각 분할 DB의 크기가 약 1GB로, 전체 약 26GB의 색인 DB를 생성하였다.

색인어로 단일어만 사용할 경우와, 단일어+복합어를 사용할 경우의 검색 성능 비교를 위하여 단일어 색인 DB를 위와 같은 방법으로 만들었다. 색인에 소요된 시간은 비슷했으며, 총 약24GB의 색인 DB가 생성되었다.

4.2 질의 처리 및 적합성 판정

질의로는 TREC-9의 50개의 Topic(Topic 451~500)를 사용하였다. 형식은 다음과 같으며, 질의 필드 중에서 <TITLE> 또는 <TITLE>+<DESCRIPTION>필드의 문장을 사용자 질의문장으로 가정하여 질의어를 추출하였다.

검색 후 Score에 따라 랭킹된 상위 1000개의 문서를 검색 결과로 사용하여 TREC-9의 적합 문서 집합과, trec_eval 프로그램을 이용하여 성능을 평가 하였다.

4.3 실험 결과

색인, 검색 과정에서 사용되는 각 파라메터는 모두 환경 파일을 통해 조정 할 수 있다. 각 파라메터는 검색 성능을 최적화 하기 위하여 여러 회에 걸쳐 반복 튜닝된다.

[표 1]은 이렇게 튜닝된 파라메터를 사용하여 얻은 최종 실험 결과이다. 유사 적합성 피드백을 사용하지 않고, 색인어와 질의어에서 모두 복합어를 생성하고, 질의어로는 Topic의 <title>만을 사용한 실험을 기준(baseline)으로 하여 각 실험에 대한 성능을 비교 하였다.

[표 1]. 평균 정확도/재현율(Ave. Precision/Recall)

	질의 확장을 하지 않은 경우			질의 확장을 한 경우			
	title		title+desc	title only			
	no phrases	phrases	phrases	phrases			
		(baseline)		식(3)	식(4)	식(5)	식(6)
Retrieved	46311	43515	43515	46311	50000	46311	50000
Relevant	2617	2617	2617	2617	2617	2617	2617
Rel_ret	1449	1470	1577	1485	1473	1441	1442
Precision	0.1740 (-0.29%)	0.1745	0.2188 (25.39%)	0.1758 (0.75%)	0.1740 (-0.29%)	0.1781 (2.06%)	0.1837 (5.27%)
R-Precision	0.1962 (1.03%)	0.1942	0.2399 (23.53%)	0.1954 (0.62%)	0.1940 (0.10%)	0.2049 (5.51%)	0.2082 (7.21%)

4.4 분석

4.4.1 단일어 v.s 복합어

[표 1]의 첫번째 열과 두 번째 열은 단일어만 색인어와 질의어로 추출했을 경우(첫번째 열), 단일어 뿐만 아니라 품사 태그 패턴에 기반하여 생성된 복합어도 색인어와 질의어로 추출했을 경우에 대한 비교이다.

후자의 경우에 검색된 문서수가 줄어들었음에도 불구하고 정확도, 재현율이 높아졌다. 이는 단일어와 복합어를 모두 사용했을 경우 상위로 검색되는 문서 수는 줄어 들지만, 검색된 문서 내에 적합 문서를 더 많이 포함할 수 있음을 보여준다. 다만 기대했던 성능향상에는 미치지 못했는데, 이는 질의로 사용한 Topic의 <title> 필드가 아주 간단한 자연어 형식이거나 또는 단순한 단어의 나열인 경우 언어 분석에 의한 복합어가 충분히 생성되지 않기 때문으로 분석된다. 실제로 TREC-9의 50개 Topic의 <title> 필드 평균 길이는 약 3.5 words에 불과하다.

4.4.1 유사 적합성 피드백

[표 1]의 4번째 열은, 식(3) TSV 함수를 사용하여 질의 확장 및 가중치 부여를 한 실험이다. 정확도에 있어 기준 실험과 비교하여 0.75%의 향상을 보였고 재현율에 있어서는 유사 적합성 피드백을 사용한 다른 경우와 비교하여 가장 높은 값을 보였다. 특히 기준 실험과 비교하여 검색된 문서의 수가 많이 증가했음을 볼 수 있다. 그러나 유사 적합성 피드백에 의한 성능향상의 정도는 미약했다. 이는 식(3)에 의해서 원 질의어 리스트에 확장되는 대부분의 키워드의 TF값이 거의 1~2로 낮기 때문으로 분석된다.

이러한 이유로 TSV함수에 tf를 반영했을 경우의 성능 향상 정도를 알아보기 위하여 식(4),(5),(6)과 같이 문서 길이로 정규화된 TSV 함수를 사용하여 실험해 보았다.

5번째 열의 식(4)를 사용한 실험에서는, 50개의 Topic이 각각 1000개의 문서를 검색하여 전체 50000개의 검색 문서 수를 기록했지만, 기준 실험에 비해 정확도는 내려가고 재현율도 거의 오르지 않았다.

6번째 열의 식(5)을 사용한 실험에서는, 위의 경우와 달리 문서 길이 또한 전체 문서 집합의 모든 문서의 평균 길이를 이용하여 정규화 하였는데, 기준 실험과 비교하여 2.06%의 정확도가 향상되었다. 특히 R-Precision은 5.51% 향상되었다.

7번째 열의 식(6)을 사용한 실험에서는, 앞의 2경우와는 달리, 기준 실험과 비교하여, 검색 문서 수, 정확도, R-Precision에서 큰 향상이 있었다. 특히 식(4)와 비교했을 때, 똑같이 50000문서를 검색 했음에도 재현율은 다소 감소했지만, 정확도가 향상되었다. 일반적으로 검색 시스템의 성능 평가에서 재현율보다 정확도를 중요시 한다는 점을 고려 한다면, 식(6)을 사용한 질의어 확장은 바람직하다고 볼 수 있다.

5. 결론 및 향후 계획

본 논문에서는 포항공과대학교 자연어 처리 연구실에서 개발한 영어 자연어 검색 시스템 POSNIR/E를 이용하여 확률 모델의 검색 모델에서의 유사 적합성 피드백에 의한 질의어 확장과 가중치 방법을 제시하고 실험을 통하여 성능 향상 정도를 분석하였다. 확률 모델에서는 벡터 모델과는 달리 유사 적합성 피드백을 통한 질의어 확장이나 가중치 부여 등 정형화된 방법이 없다. 실험에서 알 수 있듯이 TSV 함수의 형태에 따라 질의어 확장에 의한 성능 저하나 향상이 생겼다.

TSV 함수에 적합 문서 정보에 기반한 ICF 요소 뿐만 아니라, TF 요소까지 반영할 경우 유사 적합성 피드백에 의한 뚜렷한 성능향상이 있다. 그리고 TF 요소를 사용할 경우, 전체 문서의 평균 길이로 정규화 시키는 방법을 제시한다. 또한, 태그 패턴에 기반하여 생성한 복합어를 색인어, 질의어로 추출할 경우 성능 향상에 기여함을 보여준다.

본 논문에서는 TSV 함수에 문서 집합이나 질의로부터 얻을 수 있는 정보 중에서 TF, ICF요소만 반영하였다. 앞으로 여러 가지 정보를 반영하여 정교한 TSV 함수를 구성하는 연구가 계속 되어야 할 것이다. 또 의미 있는 복합어 생성을 위해 좀더 다양하고 정교한 태그 패턴을 추출하는 일이 필요하다. 본 논문의 방법으로 TREC-10 (2001년)에 4개의 런(run)을 제출했고, 결과는 TREC-9의 다른 시스템과 비교해서 3등 정도의 좋은 결과를 얻었음을 확인 할 수 있었다.

6. 참고 문헌

[1] Rocchio and J.J. (1961). Relevance Feedback in Information Retrieval. In *The Smart System-experiments in automatic document processing*, 313-323. Englewood Cliffs. NJ: Prentice Hall Inc.

[2] Ide, E. (1971). New experiments in relevance feedback. In *The Smart system-experiments in automatic document processing*, 337-354. Englewood Cliffs, NJ: Prentice Hall Inc.

[3] Robertson S.E. and Sparck Jones K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3), 129-146.

[4] Harper D.J. and Van Rijsbergen C.J. (1978). An Evaluation of Feedback in Document Retrieval Using Co-Occurrence Data. *Journal of Documentation*, 34(3), 189-216.

[5] Spark Jones K. (1979). Search Term Relevance Weighting Given Little Relevance *Information, Journal of Documentation*, 35(1), 30-48.

[6] Croft W.B. and Harper D.J. (1979). Using Probabilistic Models of Document Retrieval Without Relevance Information. *Journal of Documentation*, 35(4), 285-295.

[7] Wu H. and Salton G. (1981). The Estimation of Term Relevance Weights using Relevance Feedback. *Journal of Documentation*,37(4), 194-214.

[8] Harman D. (1988). Towards Interactive Query Expansion, Paper presented at *ACM Conference on Research and Development in Information Retrieval*. Grenoble, France.

[9] Salton G. and Buckley C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4), 288-297.

[10] Robertson S.E. et al. (1995). Okapi at TREC-3. In Overview of the Third Text Retrieval Conference(TREC-3). 109-126.

[11] Robertson S.E and Walker S. (1997). On relevance weights with little relevance information, In , *Proceeding of the 20st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 16-24.

[12] Mitra M., Singhal A. and Buckley C. (1998). Improving Automatic Query Expansion. In *Proceeding of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 206-214.

[13] Lee and Cho, (2000). Statistical Natural Language Query System THE IR based on P-Norm Model: Korean TREC-1, In *Proceeding of The 5th Korea Science & Technology Infrastructure Workshop*, 189-202.

[14] Porter, M.F. 1980. An algorithm for suffix stripping. *Program* 14:130-137.