

공학석사학위논문

강화학습을 이용한 웹 정보 검색

Web Information Search Using Reinforcement Learning

2002년 2월

서울대학교 대학원

컴퓨터공학과

정 태 진

초 록

본 논문에서는 기계 학습의 한 분류인 강화 학습 방법을 이용하여 웹 상에 존재하는 정보 중에서 특정 주제의 웹 페이지를 보다 더 정확히 찾는 방법을 제시한다.

범용 검색 엔진들은 정보소스에 접근해서 정보를 수집하는 크롤러를 이용하여 웹 링크 그래프를 방문하는데, 이들은 너비 우선 검색 방법을 사용한다. 이는 웹 페이지의 대부분을 이루는 HTML 문서의 구조적인 특징인 하이퍼링크상의 정보를 이용하는데 있어서 효과적이지 못하다. 이에 반해, 강화 학습은 웹 상의 하이퍼링크를 따라가는 문제에 있어서 미래에 이로움을 주는 행동의 효용성을 측정하는데 있어서 이점을 보인다. 이와 관련하여 웹 페이지의 하이퍼링크 정보를 분석하는 순위화 알고리즘이 많이 연구되고 있다.

제안된 방법은 200개의 인공지능 학회 관련된 웹 페이지로 이루어진 데이터 집합과 1000여 개의 웹 페이지와 7000여 개의 하이퍼링크로 이루어진 WebKB에서 제공하는 데이터 집합 등, 두 가지 데이터에 적용되었다. 범용 검색 엔진에서 사용되는 너비 우선 검색 방법과의 비교 실험을 통하여 더 적은 링크를 따라가고도 더 정확한 검색 결과를 나타내었다.

주요어: 정보 검색, 하이퍼링크, 강화 학습, 너비 우선 검색

제 목 차 례

1. 서론	1
1.1 연구 배경	1
1.2 논문의 구성	4
2. 관련 연구	5
2.1 순위화 알고리즘	6
2.2 강화학습의 응용	9
3. 정보 검색 시스템	13
3.1 불린 모델	14
3.2 벡터공간 모델	16
3.3 확률 모델	19
4. 강화학습 알고리즘을 이용한 정보 검색	21
4.1 강화학습 알고리즘	21
4.2 강화 학습을 이용한 정보 검색	27
5. 실험 및 결과	31
5.2 실험 결과 및 분석	33
6. 결론 및 향후과제	37
참고 문헌	38

1. 서론

1.1 연구 배경

인터넷상의 정보의 증가로 인해 사용자는 예전보다 더 많은 정보에 접근할 수 있는 반면 다른 한 편으로는 급속하게 늘어난 수많은 정보 중에서 자신에게 유용한 정보를 찾는 데 더 많은 시간을 투자해야만 한다. 웹 상에서 공개적으로 색인 가능한 웹 문서의 수가 25억 개를 넘었고, 동적으로 생성되는 웹 페이지 수는 5500억 개 정도이고, 하루에 새롭게 생겨나는 웹 문서도 백만 개가 넘는다[Lyman et al., 2000]. 이렇게 증가하고 있는 웹의 성장으로 인하여 웹 상에 존재하는 정보를 찾고, 이를 모아서 가져오고, 분류하는데 있어서 상당한 문제가 발생된다. 대표적인 검색 엔진인 구글(google) 시스템도 5억 6천만 개 정도의 웹 페이지를 색인해서 가지고 있을 뿐이다[Google, 2000]. 동적으로 변하는 웹의 성질 때문에 색인한 정보를 다시 검색해서 변화된 문서를 반영하는 일정한 검색 주기(refreshment rate)도 쏟아져 나오는 정보를 따라가기가 힘들 정도이다. 이런 양적인 증가도 한 원인이지만 일반적인 모든 주제에 대하여 검색을 하기 때문에 범용 검색 엔진(general-purpose search engine)을 이용하여 특정 주제에 대하여 검색을 해도 개인 사용자의 정보 요구(information need)를 정확하게 만족시키기가 힘들어지고 있다. 따라서 사용자가 특정 형태나 특정 주제(topic)에 대한 정보를 찾고자 할 때, 제한된 영역에 대해서 전문적으로 검색을 해주는 특정 영역 검색 엔진(domain-specific search engine)을 이용하면 검색의 효율성이 증가될 수 있다. 이 검색 엔진의 장점은 특정 주제에 관련된 질의 검색의 결과의 정확도가 범용 검색 엔진에 비하여 높다[McCallum et al., 1999].

다음은 특정 영역 검색 엔진들에 대한 예를 나열하고 있다.

검색 엔진 이름	설 명
www.campsearch.com	아이들과 성인들을 위한 여름 캠프에 대하여 검색을 해준다. 사용자는 지역적인
www.netpart.net	위험차별회사, 시간회, 로시, 주, 생활이용, 회사. 회사의 웹 페이지를 검색할 수 있다.
www.mrqe.com	영화의 비평을 영화 제목으로 검색할 수 있다. 또한 신문, 잡지에 실린 관련 비평에
www.maths.usyd.edu.au	대학에서 출판한 웹 페이지를 검색 할 수 있다.
www.travel-finder.com	여행에 대한 웹 페이지를 카테고리, 위치, 활동 등에 대한 정보로 검색할 수 있다.

표 1: 특정 영역 검색 엔진의 예

웹 상의 문서를 검색하는 효율적인 방법으로 웹 문서의 대부분을 이루는 HTML 문서의 구조를 이해하고 이에 관련된 정보를 바탕으로 검색의 효율을 향상시킬 수 있다. HTML은 다음과 같이 일반 텍스트 문서가 가지고 있지 않은 두 종류의 구조로 구성되어 있다[Boyan et al., 1996].

첫째는 내부적인 구조로, 이는 HTML 태그에 의해 표시된 텍스트 조각으로 구성된다. HTML은 문서의 텍스트에 대해 일정한 역할을 부여하는데, 어떤 것은 볼드체나 이탤릭체 등과 같은 포맷과 관련되어 있으며, 헤드라인이나 앵커(anchor), 다른 문서와의 연결을 위한 하이퍼링크(hyperlink) 부분에 해당하는 텍스트 조각과 같이 보다 풍부한 의미를 내포하게 하는 역할도 있다.

둘째는 외부적인 구조이다. 하이퍼텍스트(hypertext)상의 한 노드로써,

HTML 페이지는 무수히 많은 다른 페이지들과 잠재적으로 연결되어 있다. 이것은 하이퍼링크를 통해 연결되어 있으며, 자신이 링크하고 있는 다른 페이지와 자기 페이지를 링크하고 있는 다른 페이지 등으로 서로 얽혀 있다.

본 논문에서는 기존의 웹 검색 엔진에서 사용되어지는 정보를 검색하는 너비 우선 검색 방법의 단점을 지적하고, 웹 페이지의 중요한 특징인 하이퍼링크로 이루어진 구조에 대한 분석을 통하여 웹 상을 돌아다니면서 관련 정보를 검색하는 일종의 에이전트인 크롤러(crawler)를 기계학습(machine learning)의 한 종류인 강화학습(reinforcement learning)을 이용하여 지능적이고 효율적인 검색을 하는 방법을 제안하고 실험을 통하여 이를 알아보겠다.

1.2 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 웹 문서 검색에 관련된 순위화 알고리즘에 대한 연구와 정보 검색에 강화 학습을 이용한 연구에 대해 설명한다. 3장에서는 웹 문서 검색을 위해 사용한 정보 검색 시스템에 대해 기술하며 4장에서는 강화 학습의 이론적인 설명과 이를 이용하여 정보 검색의 효율성을 증가시키기 위하여 제안한 강화 학습에 바탕을 둔 정보 검색에 대하여 기술한다. 5장에서는 한 개의 인공 데이터 집합과 카네기 멜론 대학의 연구실에서 제공하는 WebKB 데이터를 가지고 한 실험 결과에 대하여 기술한다. 마지막으로 6장에서는 연구 내용을 요약하고 앞으로의 연구 과제와 함께 결론을 맺는다.

2. 관련 연구

웹으로부터 효율적으로 정보를 수집하는 웹 검색 알고리즘이 많이 개발 향상되어왔다. 일반적인 범용 검색 엔진에서 사용하는 너비 우선 (breadth-first-search) 검색의 단점을 보완하기 위해서 웹의 구조 즉, 하이퍼 링크와 관련된 정보를 이용하는 연구가 많이 이루어졌다. 크게 보면 다음의 3가지의 연구로 분류될 수 있는데, 본 장에서는 링크 정보를 이용한 순위화 방법과 강화학습을 이용한 방법에 대하여 기술한다.

첫째, 앞서 말한 설명했듯이 검색의 영역을 줄이는 대신 사용자의 정보 요구를 더 정확히 찾아 주는 특정 영역 검색 엔진에 대한 연구가 진행중이다.

둘째, 웹 문서를 정확히 검색하고 분류하기 위해 링크와 링크의 순서를 이용하는 순위화 방법이 연구되었다[Chakrabarti et al., 1997]. 이 연구에서 그들은 문서 내에서 한 링크에 가까이 위치한 링크의 경우, 같은 주제를 가진 페이지를 가리키고 있을 가능성이 높다는 것을 보였다.

셋째, 마지막으로 강화학습을 이용한 정보 검색의 연구가 있다. 이는 강화학습의 프레임 워크를 이용하여 보상값을 근사화하는 방법에 따라 다르게 나타날 수 있다.

2.1 순위화 알고리즘

대부분의 탐색 엔진들은 순위화를 위하여 불리안과 벡터 공간 모델의 변형을 이용한다. 탐색과 마찬가지로 순위화는 텍스트를 접근하지 않고 색인만 사용하여 수행되어야 한다. 현재 탐색 엔진들이 사용하는 특정 순위화 알고리즘에 관한 공유 정보는 많지 않다. 더구나, 다른 탐색 엔진들을 공정하게 비교해서 그것들의 차이점을 부여하거나 개선을 계속하는 것도 어렵다. 더 중요한 것은 재현율(recall)을 측정하는 것이 불가능하다는 것인데, 그 이유는 관련 페이지의 수가 간단한 질의에 대하여 매우 크기 때문이다.

기존의 tf-idf방식 외에 세 가지 순위화 알고리즘을 제안하고 있는 연구가 있었다[Yuwono and Lee, 1996]. 이들은 Boolean spread, vector spread, most-cited라고 불린다. 처음 두 개는 응답내의 한 페이지가 가리키는 페이지들 혹은, 응답내의 한 페이지를 가리키는 페이지들을 포함하기 위해 확장된 불리안과 벡터 모델의 정규 순위화 알고리즘들이다. 세 번째에 있는 most-cited는 응답내의 페이지들에 대한 링크를 갖고 있는 페이지들에 포함된 용어들에 기반 한다. 2400개의 웹 페이지들로 구성된 컬렉션에 대하여 56개의 질의들을 고려하여 이들 기법들을 비교하면 벡터 모델이 평균 75%의 정확률을 가진 더 좋은 결과를 보인다.

새로운 순위화 알고리즘의 몇몇은 하이퍼링크 정보를 이용하는데, 이는 웹과 정상적인 IR 데이터베이스 간의 주요 차이점이다. 한 페이지를 가리키는 하이퍼링크의 수는 그것의 평판도(popularity)와 질(quality)의 수준을 제공한다. 또한 페이지들에 공통적인 많은 링크들 혹은 동일 페이지에 의해 참조된 페이지

들은 가끔 그 페이지들 사이의 관계를 나타낸다. 여기에 이 사실들을 활용한 순위화 기법의 3가지 예제를 제시한다. 그러나 이들 중 둘은 질의에 의존하고 나머지 하나는 그렇지 않다는 점에서 서로 차이가 있다.

첫 번째는 WebQuery[Carriere and Kazman, 1997]인데, 웹 페이지들의 시각적 표시를 허락한다. 이 시스템은 질의에 대한 응답이 웹 페이지의 집합을 취하여 각 웹 페이지가 어떻게 연결되는냐에 따라서 이들을 순위화한다. 부가적으로 이것은 본래의 집합에 강하게 연결된 웹 페이지들을 찾아 그 집합을 확장하기도 한다[Li, 1998].

둘째로, 더 좋은 방법이 Kleinberg에 의하여 제안되었다[Kleinberg, 1998]. 이 순위화 방법은 질의에 의존하며, 응답에 있는 페이지들 가리키거나 그 페이지들이 가리키는 페이지들의 집합 S 를 고려한다. S 에서 자신에게 들어오는 많은 링크들을 갖고 있는 페이지들은 전거(authorities)라 불린다. 이들끼리는 연관 정보를 갖고 있다고 생각 할 수 있다. 나가는 링크들을 많이 갖고 있는 페이지들은 허브(hub)라 불린다. 이들은 비슷한 내용을 가리킬 경우 많다. 긍정적인 양 방향 피드백은 더 좋은 전거는 좋은 허브들로부터 들어오는 페이지들에서 오고, 더 좋은 허브 페이지는 좋은 전거들로 나가는 페이지에서 온다. $H(p)$ 와 $A(p)$ 를 페이지 p 의 허브와 전거 값이라 하면, 이 값들은 다음 식의 모든 페이지들 p 에 대하여 만족되도록 정의된다.

$$H(p) = \sum_{u \in S \mid p \rightarrow u} A(u) , \quad A(p) = \sum_{v \in S \mid v \rightarrow p} H(v) \quad (1)$$

여기서 모든 페이지들에 대한 $H(p)$ 와 $A(p)$ 는 정규화 된다.

셋째로, Google 검색엔진에서 사용한 페이지랭크(PageRank)이다[Brin and Page, 1998]. 이는 확률 q 를 가지고 임의의 페이지로 넘어가거나, 확률 $1-q$ 를 가지고 현재 페이지 상에서 임의의 하이퍼링크를 따라가는 것과 같이, 웹에서 임의로 항해하는 사용자를 모의실험하고 있다. 여기에는 사용자가 이미 탐색된 하이퍼링크의 후방을 따라 앞서 방문한 페이지로 돌아가지 않는다는 가정을 하고 있다. 이 프로세스는 마코프 체인으로 모델화될 수 있으며, 각 페이지에 있는 고정 확률이 계산되어 구글의 순위화 메커니즘의 부분으로 사용된다. $C(a)$ 가 페이지 a 의 나가는 링크들의 수라고 하고, 페이지 a 가 p_1 에서 p_n 까지의 페이지들에 의해 가리켜진다고 가정하자. 그러면 페이지랭크 즉, a 의 $PR(a)$ 는 아래 식 (2)와 같이 정의된다.

$$PR(a) = q + (1-q) \sum_{i=1}^n PR(p_i) / C(P_i) \quad (2)$$

여기서 q 는 시스템에 의해 설정되어야 한다. 다른 페이지들의 가중치는 해당 페이지 안의 링크 수에 의해 정규화 된다. 페이지랭크는 반복 알고리즘을 사용하여 계산될 수 있으며, 웹의 정규화된 링크배열의 고유벡터에 대응한다. 이 순위화를 사용하여 웹에서 정보를 검색하는 것이 다른 기법보다도 더 좋은 것으로 나타났다.

2.2 강화학습의 응용

강화 학습이 응용된 사례는 여러 분야에 걸쳐 상당히 많으나 이 중 정보 검색분야에 적용된 예로는 LASER 시스템이 대표적이다[Boyan et al., 1996]. 라이코스나 알타비스타와 같은 웹 검색 엔진들은 웹 상에서 필요한 정보를 찾기 위한 중요한 도구가 되었다. 이러한 검색 엔진들은 사용자가 원하는 문서를 찾기 위해 보통 통계적 방법을 사용한다. 한편 검색 엔진의 입장에서 간과해서는 안 될 부분이 있다면, 그것은 대부분의 웹 문서들이 HTML의 형식을 취하고 있다는 점일 것이다. 검색 엔진은 HTML 문서가 가지는 구조 그 자체를 분석하여 통계적 방법에 그대로 이용할 수 있기 때문이다. HTML 문서는 일반 문서와는 달리 다음의 두 가지 구조를 지니고 있다. 첫째, HTML 문서는 태그와 포맷에 의해 그 내부적 구조가 결정된다. 특히 태그와 포맷은 문서 내의 단어 혹은 문장의 중요도를 결정할 수 있다. 예를 들어, 굵은 글씨로 쓰여 있는 단어나 title 태그내의 문장은 문서 내 일반 단어나 문장에 비해 더 중요하다고 고려될 수 있다. 둘째, HTML 문서는 하이퍼링크에 의해 그 외부적 구조가 결정된다. HTML 페이지들은 그 페이지들이 포함하는 하이퍼링크나 그 페이지들을 가리키는 하이퍼링크를 통해 잠재적으로 관련된다. 이렇듯 HTML 문서들은 일반 문서들에 비해 구조화가 더 잘 되어 있기 때문에, 웹 검색 엔진들은 전통적인 통계적 검색 방식에 HTML 구조를 이용한 기법들을 결합함으로써 그 성능을 향상시킬 수 있다.

LASER는 기계 학습 기법을 적용하여 웹 페이지들을 색인하는 시스템이다. 사용자의 관점에서 보자면, LASER는 일반 검색 엔진과 크게 다를 바가 없다. 사용자가 LASER에 질의를 주면, LASER는 그 질의와 가장 관련이 있다고 판단되는 60개의 페이지들에 대한 링크와 그 페이지들의 초록(abstract)을 링크의

랭킹에 따라 보여준다. LASER의 검색 함수(retrieval function)는 tf-idf 벡터 공간 검색 모델 (vector space model)에 기반한다[Salton, 1989]. 이 모델에서 문서와 질의는 실수값들의 벡터로 표현되며, 각각의 실수값들은 tf-idf 값에 따라 문서와 질의 내의 각각의 단어의 가중치(weight)를 표시한다. 문서와 질의는 결국 같은 차원의 벡터로 표현되는데, LASER는 문서와 질의 사이의 관련성이 두 벡터의 내적(inner product)값을 계산하여 평가한다. LASER의 검색 함수는 랭킹에 영향을 주는 여러 개의 파라미터들로 구성된다. 파라미터 값들에 의해, 헤드라인(headline)과 같은 어떤 HTML 부분에 위치한 단어들의 가중치를 크게 할 것인가, 하이퍼링크는 랭킹을 결정하는데 어느 정도 반영할 것인가, 단어 전체가 아닌 일부분만 일치하는 경우 이를 어떻게 해석할 것인가 등등이 결정될 수 있다. LASER는 총 18개의 파라미터를 가진다. 파라미터 값들을 특정한 값으로 결정하면, 그에 따라 수많은 검색 함수들 중 특정한 검색 함수가 얻어진다. 검색 함수는 이렇게 파라미터 값들을 조정해 나감으로써 특정 사용자나 특정 문서 집합에 대해 적용해 나갈 수 있다. 대부분의 웹 페이지들은 HTML로 쓰여져 있다. HTML은 웹 페이지의 디자이너가 문서의 특정 부분에 어떠한 의미를 부여하도록 허용해 준다. 예를 들어, 디자이너는 문서의 제목(title), 헤드라인, 하이퍼링크, 굵은 글씨체, 이탤릭체 등의 문자 포맷 등을 HTML 문서상에 기술할 수 있다. LASER는 이러한 HTML 문서의 구조를 이용한다. 예를 들면, LASER의 한 파라미터는 문서의 타이틀이 문서내의 다른 단어들에 비해 얼마나 비중 있는지를 결정한다. 이러한 파라미터들은 그와 관련된 단어의 tf-idf 값에 곱해져 그 단어의 가중치를 상대적으로 높이거나 낮춘다.

구체적으로 설명하자면, LASER는 검색에 있어 훌륭한 시작 지점이 될만한 페이지들에 높은 랭크(rank)를 부여한다. 이에 따라 LASER는, 어떤 한 페이지가 질의와 관련된 정보를 담고 있지 않아도 다른 많은 관련된 문서들을 링크하고 있다면, 그 페이지는 질의와 관련이 있다고 추정한다. 같은 맥락에서, 에이전

트는 당장은 관련된 문서를 얻을 수 없어도 그 문서를 거쳐 그 문서내의 하이퍼링크들을 따라가다 보면 관련된 문서들의 클러스터(cluster)를 발견할 수 있도록 학습된다. 이렇게 함으로써 에이전트는 전체적으로 그가 받는 보상을 최대화할 수 있다. 다음은 페이지에 랭크를 부여하는 식을 제시하고 있다.

$$rsv_{t+1}(q, d) = rsv_0(q, d) + \gamma \sum_{d' \in \text{links}(d)} \frac{rsv_t(q, d')}{|\text{links}(d)|^\nu} \quad (3)$$

여기서 γ 는 이웃한 페이지들의 영향을 조절하는 디스카운트 팩터(discount factor)이다. 그리고 $\text{links}(d)$ 는 페이지 d 내의 하이퍼링크들에 의해 링크되는 페이지들의 집합이다. ν 는 정규화를 조절하는 파라미터이다. 이 다이내믹 프로그래밍 업데이트(dynamic programming update)는 문서들의 전체 집합의 한 부분 집합 내의 각각의 문서들에 대해서 반복적으로 수행된다. 이 부분 집합에는 의미가 있는 rsv_0 값을 가지는 문서들뿐만 아니라 이러한 문서들로 링크되는 문서들까지 포함된다. 후자의 문서들을 포함하는 이유는, 그 문서들 자체가 질의와 관련된 정보를 담고 있지 않아도 질의와 관련된 문서들을 링크하고 있으므로, 그 문서들 또한 검색의 훌륭한 시작 지점으로 고려될 수 있기 때문이다.

실제 5번 정도의 업데이트를 통해 각 문서 d 에 대한 $rsv(q, d)$ 의 값은 수렴된다. 바로 이 수렴된 값이 문서 d 의 하이퍼링크와 관련된 랭킹 점수가 되어, 질의 q 에 대한 문서 d 의 총 랭킹 점수에 기여하게 된다.

또 다른 강화학습을 이용한 예로는 카네기 멜론 대학에서 연구한 웹 와쳐(WebWatcher)시스템이 있다[Joachims et al., 1997]. 이 시스템은 웹 환경에서 사용자의 브라우징을 도와주는 에이전트 소프트웨어이다. 웹 브라우징 에이전트인 웹 와쳐는 박물관 안내원처럼 가이드 역할을 해준다. 즉, 사용자에게 적절

한 브라우징 경로를 제공해준다. 그런데, 이 경로는 사용자의 관심사와 기존의 사용자들이 브라우징한 정보에 기반 하여 작성되며 그 유효성도 검증된다. 웹 와쳐는 기존 사용자들의 브라우징 시퀀스로부터 각 하이퍼링크를 태깅한다. 즉 사용자가 하이퍼링크를 클릭하면 하이퍼링크 태깅 정보가 확장된다. 사용자에게 가이드 역할을 하기 위해 이 시스템은 사용자의 관심사를 현재 페이지에 있는 모든 하이퍼링크의 태깅 정보와 비교를 하여 사용자의 관심사와 가장 유사한 태깅 정보를 지닌 하이퍼링크를 추천한다. 사용자가 자신이 원하는 정보를 찾기 위해서 하이퍼링크를 선택하는데 있어서 추천을 하기 위하여 강화학습 알고리즘을 이용하여 우선 순위를 결정한다.

3. 정보 검색 시스템

정보 검색 시스템에서 정보 자료를 검색하기 위해서 사용되는 기법을 검색 기법이라 한다. 그리고, 정보 검색 시스템에서 주어지는 일정한 형식의 질의어를 검색문이라고 한다. 정보 검색 기법으로 사용하는 전통적인 방법에는 다음과 같은 것들이 있다.

- 불린 모델(boolean model)
- 벡터공간 모델(vector space model)
- 확률 모델(probabilistic model)

불린 모델은 집합 이론에 근거하고 있으면, 질의와 문서가 색인어의 집합으로 표현된다. 벡터 공간 모델은 질의와 문서가 벡터로 표시되어 n 차원 공간 상에서의 거리를 기준으로 검색을 한다. 확률 모델은 확률론에 근거해 질의와 문서와의 관계를 확률적으로 계산한다. 이러한 전통적인 검색 방법 외에도 여러 가지 모델이 제안되어 왔는데 퍼지(fuzzy) 모델, 확장 불린(extended boolean) 모델, 추론 네트워크(inference network) 모델, 신념망(belief network) 모델 등이 이에 해당한다. 이 장에서는 앞서 말한 기본적인 3가지 모델에 대하여 살펴본다.

3.1 불린 모델

불린 검색 모델은 시스템으로 구현하기 쉽고 질의를 처리하는 시간이 매우 빠르기 때문에 상용 검색 시스템에서 널리 사용되어 왔다. 불린 검색 모델에서 각 문서는 색인어의 집합으로 표현되고 질의어는 불린 수식으로 표현된다. 불린 수식은 불린 연산자 AND, OR, NOT으로 연결된 색인어들로 구성되며 시스템은 질의에 해당하는 불린 연산식을 만족시키는 문서들을 검색한다.

그러나 불린 검색 모델은 사용자의 질의에 대한 문서의 유사도(similarity)를 계산하지 않기 때문에 문서 정렬 기능을 제공하지 못한다. 따라서 이를 보강한 확장 불린 모델이 등장하였다. 불린 검색 시스템에 문서의 유사도에 따른 정렬 기능을 제공하기 위하여 MIN과 MAX와 같은 퍼지 집합 연산자를 사용하는 퍼지 집합 모델이 개발되었다. 그러나 MIN과 MAX 연산자가 질의와 문서의 유사도 계산에서 사용자의 생각과는 일치하지 않는 결과를 생성하는 문제가 발생되었고, 이를 극복하기 위하여 여러 확장 모델들이 제안되었다[Salton et al., 1983][Turtle and Croft, 1991][Callan et al., 1992]. 이 모델들은 AND와 OR에 대하여 서로 다른 연산자들을 사용하지만 확장된 불린 모델을 기반으로 하여 설명될 수 있다.

확장 불린 검색 방법의 하나인 P-norm 모델은 역시 불린 모델의 단점을 개선하기 위해 제안된 것으로, 색인어에 대한 가중치뿐만 아니라 질의어에 대한 가중치도 갖는다. P-norm 모델에서는 용어 A_1, A_2, \dots, A_n 에 대한 색인 가중치 $d_{A_1}, d_{A_2}, \dots, d_{A_n}$ 을 갖고 있는 문서 D 를 n 차원 공간에 있는 $(d_{A_1}, d_{A_2}, \dots, d_{A_n})$ 좌표에 있는 것처럼 처리한다. $d_{A_1} \text{ OR } d_{A_2} \text{ OR } \dots \text{ OR } d_{A_n}$ 형태의 OR 질의가 있다고 가정하면 이 질의에 대해 모든 n 좌표가 0인 문서는 검색에서 제

의된다. $d_{A1} \text{ AND } d_{A2} \text{ AND } \dots \text{ AND } d_{An}$ 형태의 AND 질의에 대해서는 모든 n 좌표가 1인 문서, 즉 모든 색인 용어 가중치가 1인 문서가 질의에 대한 결과로 가장 적합하다고 결정된다. OR 질의에 대해서는 $(0, 0, \dots, 0)$ 좌표에서 떨어진 거리를 내림차순으로, AND 질의에 대해서는 $(1, 1, \dots, 1)$ 좌표에서 떨어진 거리를 오름차순으로 정렬해서 문서 가중치를 부여한다.

P-norm 모델에서 사용하는 일반적인 질의 형태는 식 (4)과 같다. 상관 계수 p 는 연산에 대한 제한 정도를 나타낸다.

$$Q_{OR_p} = (A_1, a_1) OR_p (A_2, a_2) OR_p \dots OR_p (A_n, a_n) \quad (4)$$

$$Q_{AND_p} = (A_1, a_1) AND_p (A_2, a_2) AND_p \dots AND_p (A_n, a_n)$$

또한 질의 용어에도 상대적인 중요도에 따라 가중치 a_i 를 부여한다. P-norm 모델에 대한 질의와 문서 사이의 유사도는 식 (5)를 사용해서 계산한다.

$$\text{SIM}(Q_{or_p}, D) = \sqrt[p]{\frac{a_1^p d_{A1}^p + a_2^p d_{A2}^p + \dots + a_n^p d_{An}^p}{a_1^p + a_2^p + \dots + a_n^p}}$$

$$\text{SIM}(Q_{and_p}, D) = 1 - \sqrt[p]{\frac{a_1^p (1 - d_{A1})^p + a_2^p (1 - d_{A2})^p + \dots + a_n^p (1 - d_{An})^p}{a_1^p + a_2^p + \dots + a_n^p}} \quad (5)$$

$$\text{SIM}(Q_{not}, D) = 1 - \text{SIM}(Q, D)$$

3.2 벡터공간 모델

표준 불린 모델은 질의나 문서의 키워드에 모두 이진(binary) 가중치를 할당하는데 비해 벡터공간 모델은 질의나 문서의 키워드에 이진 값이 아닌 적절한 가중치를 할당할 수 있다. 또한 불린 모델에서 할 수 없는 부분 매칭이 가능하며, 따라서 질의와 문서의 유사도에 따라 랭킹을 줄 수 있다. 벡터 공간 모델에서 모든 색인어는 서로 독립이라고 가정을 하며 질의와 문서는 모두 식 (6)과 같은 벡터로 표현된다.

$$d_i = (w_{i1}, w_{i2}, \dots, w_{in}) \quad (6)$$

위 표현에서 d_i 는 문서 또는 질의를 표현하며, w_{ik} 는 문서 d_i 에서의 색인어 t_k 에 대한 가중치 값이다. 문서 i 에 나타나지 않는 색인어에는 가중치 0이 할당된다. 문서 또는 질의 벡터들이 형성된 이후의 검색과정은 벡터의 연산에 의해 이루어진다. 문서 d 가 $(w_{i1}, w_{i2}, \dots, w_{in})$ 로 표현되고 질의 q 가 $(w_{q1}, w_{q2}, \dots, w_{qn})$ 로 표현되었을 때, 문서 d 와 질의 q 사이의 벡터 유사도 측정은 두 벡터 \vec{d}_i 와 \vec{q} 사이의 상관도로 구할 수 있으며, 이는 두 벡터간 사이 각의 코사인 값으로 계산될 수 있다. 이는 아래와 같은 수식으로 나타낼 수 있으며, 이를 코사인 유사도(cosine coefficient similarity)라고 한다.

$$\text{sim}(d_i, q) = \frac{\vec{d}_i \cdot \vec{q}}{|\vec{d}_i| |\vec{q}|} \quad (7)$$

가중치 w_{ik} 가 0보다 크거나 같은 값을 갖기 때문에 $\text{sim}(d_i, q)$ 값은 0과 1사이의 값이 된다. 따라서, 벡터공간 모델은 문서가 질의와의 관련 여부만을 예

측하기보다는 질의와의 유사도 값에 따라 순위를 매기기 때문에, 부분적으로 질의에 정합 되는 문서라도 검색되며 또, 일정한 유사도 값 이상의 문서만을 검색하기 위하여 $\text{sim}(d_i, q)$ 값에 임계값을 둘 수도 있다.

질의와 문서의 유사도 측정에 있어서 문서 값은 색인어들의 가중치에 의해 결정되기 때문에 가중치 부여 기법은 검색 효과에 영향을 미치는 중요한 요소이다. 정보 검색에 관한 많은 연구들은 색인어에 가중치를 부여하기 위하여 출현빈도(term frequency), 문서빈도(document frequency), 정규화(document length normalization)의 세 가지 요소를 고려한다. 출현빈도는 문서 내에서 자주 출현하는 색인어에 보다 높은 가중치를 부여한다. 문서빈도는 전체 문서들 중에서 적은 수의 문서에 출현하는 색인어에 보다 높은 가중치를 부여한다. 그리고 정규화는 문서 집합에 있는 모든 벡터들의 길이를 같게 만드는 것으로써 작은 크기의 문서가 문서 값 계산에 있어서 불공평하게 취급되는 것을 피하도록 한다.

벡터공간 모델에서 사용하고 있는 벡터 원소의 가중치를 결정하는 *tf-idf* 방법을 설명하면 다음과 같다.

벡터 공간의 각각의 차원(dimension)은 각 단어에 해당하는 가중치를 나타낸다. 벡터의 각 값은(vector element) 각 문서에 대한 단어 빈도수(term frequency) $TF(w, d)$ 와 문서 빈도수(document frequency) $DF(w)$ 의 조합으로 계산된다. $TF(w, d)$ 는 단어(word) w 가 문서(document) d 에 나타난 횟수를 $DF(w)$ 는 단어 w 가 한 번 이상 나온 문서의 수를 나타낸다. 문서 빈도수로부터 역 문서 빈도수(inverse document frequency) $IDF(w)$ 를 다음의 식과 같이 계산한다.

$$IDF(w) = \log \frac{|D|}{DF(w)} \quad (8)$$

$|D|$ 는 문서의 총 개수이고 벡터의 i 번째 값 $d^{(i)}$ 는 다음과 같이 두 값의 곱으로 계산된다.

$$d^{(i)} = TF(w_i, d) \times IDF(w_i) \quad (9)$$

3.3 확률 모델

확률 모델의 1960년대 초에 Maron과 Kuhns가 확률 검색의 개념을 처음 제시한 이래 많은 연구가 진행되어 왔다[Maron and Kuhns, 1960]. 특히 1970년대 후반에 들어서는 확률 검색이 하나의 강력한 정보 검색 모델로 대두되기에 이르렀으며 현재 주로 사용되고 있는 모델은 Robertson에 의해 정립되었다 [Robertson and Sparck, 1976].

확률 모델을 개념적으로 설명하면 특정한 질문에 대해 각 문서가 관련이 있을 확률과 관련이 없을 확률을 산출하여 관련 확률이 관련이 없을 확률보다 큰 문서를 검색하는 것이다. 확률검색의 기본적인 가정은 적합성 정보가 미리 준비되어 있다는 것이다. 다시 말해 적합 문서와 부적합 문서에 대한 결정이 전체 문서집합에 대해서나 또는 일부 집합에 대해 검색 이전에 내려져 있다는 것이다.

대부분의 확률 모델은 적합 문서와 부적합 문서에서의 색인어 출현정보에 기초하고 있다. 각 문서 X 가 (10)와 같은 색인어 벡터로 표현되어 있다면, x_i 는 i 번째 색인어의 유무를 나타내는 것으로 $x_i = 0$ 또는 1의 값을 갖게 된다.

$$X = (x_1, x_2, \dots, x_n) \quad (10)$$

이때 문서가 질의를 만족하는 정도를 계산하는 유사도함수 $g(X)$ 는 다음 식 (11)과 같다.

$$g(X) = \log \frac{\Pr(X|rel)}{\Pr(X|nonrel)} \quad (11)$$

위 식에서 $\Pr(X|rel)$ 은 적합 문서가 문서 벡터 X 로 표현될 확률이며, $\Pr(X|nonrel)$ 은 비적합 문서가 문서 벡터 X 로 표현될 확률이다. 다시 말해, 문서 X 가 질의를 만족시킬 가능성이 클수록 $g(X)$ 의 값도 증가한다. 여기에서 유사도함수 $g(X)$ 의 값을 계산하기 위해서는 $\Pr(X|rel)$ 과 $\Pr(X|nonrel)$ 의 값을 추정해야 한다.

확률 검색은 이론적으로는 아주 훌륭한 검색 모델이지만, 실제로는 적합성 정보를 사전에 확보하는 것이 그리 간단하지 않기 때문에 사용상에 어려움이 많다. 그러나, 일단 검색된 문서의 적합성 판정 결과를 이용하여 검색된 문서의 순위를 매겨 줌으로써 사용자의 요구에 가장 적합한 문서를 먼저 제공할 수 있다는 장점이 있다.

4. 강화학습 알고리즘을 이용한 정보 검색

4.1 강화학습 알고리즘

기계학습을 크게 분류하면 명시적으로 학습 목표가 주어지는 분류(classification), 예측(prediction)등과 같은 감독학습(supervised learning)과 명확하게 학습의 목표가 주어지지 않는 군집화(clustering)와 같은 무감독 학습(unsupervised learning)으로 나누어 볼 수 있는데, 강화학습은 그 중간적인 특성을 띠고 있다. 강화 학습의 특성은 주어진 환경과의 상호 작용에 의한 학습(Learning from interaction with environment), 지연되는 보상(delayed reward), 그리고 시도와 오류(trial and error)로 기술할 수 있다[Sutton and Barto, 1998]. 이러한 특성을 학습 과정으로 표현하면 다음과 같다.

- 시각 t 에 학습자가 선택 가능한 행동 집합과 학습자의 행동에 따른 가설공간의 가설이 주어진다.
- 행동 집합에서 이제까지 학습한 지식을 토대로 가장 좋은 결과를 유도할 만한 행동을 선택한다.
- 특정 행동 선택에 따른 가설을 선택하고 이에 따라 가설공간을 탐색한다.
- 환경으로부터 행동에 대한 평가를 받는다(evaluative feedback).
- 학습한 지식과 평가간의 차이를 고려하며 시각이 증가한다.

강화 학습은 위에서 기술한 특성이외에 감독학습의 일부 단점을 보완해줄 수 있는 특성이 있다. 이것은 추측으로부터 추측으로 일컬어지는 특성으로 학습자

의 훈련 시 발생하는 경험을 보존할 수 있게 해준다. 감독 학습은 감독자로부터 제시한 정형화한 사례나 지식으로부터 훈련을 통해 주어진 환경을 학습하는 형태를 말하며 정형화가 가능한 정적인 특성의 환경에 주로 응용되며 특정 시스템의 동적인 상태를 표현해야 하는 응용에는 적절하지 못하다. 대부분의 실제 응용은 이러한 동적인 환경과 각 학습 시스템과의 상호 작용이 고려되어야 하는 특성을 가지고 있는데 강화 학습은 그 특성상 이러한 응용에 적합하다.

강화 학습의 역사적인 배경을 살펴보면 크게 두 가지의 연구 방향이 그 맥을 이어 오고 있는 것을 알 수 있다. 동물 행동 심리학이 그 이론의 출발지인 시행과 오류가 한 방향이며, 가치 함수(value function)와 동적 프로그래밍(dynamic programming)을 이용한 최적 제어 이론(theory of optimal control)이 또 다른 연구 방향이었다. 첫 번째의 연구 방향은 대부분의 고등동물이 자신의 환경에 적응하기 위한 학습 방식에서 유추된 것이고 두 번째의 연구방향은 주로 1950년대 중반 Richard Bellman의 Bellman 방정식과 19세기의 Hamilton과 Jacobi의 최적화 이론을 확장시켜 이론의 주요한 근거로 삼고 있다. Bellman이 제시한 Bellman 방정식과 MDP(Markov decision process)로 알려져 있는 이산적(discrete)이고 확률적인(stochastic) 특성을 갖는 모델을 최적화하는 최적 제어 문제는 현재에도 강화 학습의 중요한 이론적 바탕을 이룬다. 강화 학습은 주로 제어와 예측의 문제에 많이 적용되어 왔다. 강화 학습은 학습자가 주어진 환경과의 상호작용을 통해 자신의 학습 목표를 달성하는 실시간 학습으로 크게 다음의 4 가지의 요소로 구성된다.

4.1.1 행동 선택 정책(Policy)

정책은 강화 학습의 학습자가 특정 시간에 주어진 상태 집합에서 특정 상태

를 선택하는 행동의 기준을 말하며, 확률적인 시행을 반복하여 이를 통해 얻어진 경험, 즉 시행착오를 통해 얻어진다. 주어진 상태 집합에서 행동을 선택하는 방법은 다음과 같다.

• 탐욕적 행동 선택 정책(greedy action selection): 주어진 시각 t 에서 추정된 행동의 값들이 가장 높은 값을 갖는 행동을 선택하는 방법을 말하며 이 경우 학습자는 현재의 지식만을 이용하게 된다(exploitation).

$$Q_t(a^*) = \arg \max_a Q_t(a) \quad (12)$$

위 식에서 a^* 는 주어진 시각 t 에 최적의 행동이다.

• ϵ -탐욕 행동 선택 정책(ϵ -greedy action selection): 대부분의 행동 선택은 현재 상태에서 최적이라고 생각되는 행동(greedy action)을 선택하고 ϵ 만큼은 미래에 더 최적이라고 추정되는 행동을 선택하는 방법을 말한다. 이 경우 최적의 행동을 선택할 확률은 $1 - \epsilon$ 보다 크게 수렴한다. 즉, ϵ 만큼은 계속적으로 탐험(exploration)을 하므로 greedy 행동 선택방법보다 좀 더 빨리 최적의 목표에 도달하게 된다.

• 소프트맥스 행동 선택 정책(softmax action selection): ϵ -greedy 행동 선택방법은 탐험과 탐색의 균형을 유지하며 행동을 선택하는 효과적인 방법이긴 하지만 선택 가능한 모든 행동을 동일한 확률로 선택하는 단점이 있다. 소프트맥스 행동 선택방법은 현재 최적이라고 생각되는 행동의 선택 가능성은 최대로 유지하고 나머지 행동에 대해서는 추정된 값에 따라 가중치를 부여하여 선택 확률을 조절한다. 이를 위해 일반적으로 Gibbs 또는 Boltzmann 분포를 사용한다.

4.1.2 환경 모델(Model of the Environment)

학습자가 주어진 환경과 상호 작용을 할 때 상태(state), 행동(action), 보상(reward)이라는 세 가지 기본 틀을 이용한다. 환경은 주로 상태로 표현되며 학습자는 적절한 정책에 따라 행동을 취하게 된다. 이 때, 환경은 학습자에게 행동에 대한 보상을 주게 된다. 아래 그림은 강화 학습 에이전트가 t 시각에 행동 a_t 를 취하면 행동에 대한 보상 r_t 가 환경으로부터 주어진다. 그리고 행동에 의해서 상태 s_t 가 s_{t+1} 로 변화된다. 강화 학습에서의 환경 모델은 마코프 속성을 만족하는 MDP(Markov Decision Process)이다. 마코프 속성이란 시각 $t+1$ 의 환경에서의 반응은 오직 시각 t 에서의 상태와 행동만에 의존하는 속성을 말한다[Mitchell, 1997]].

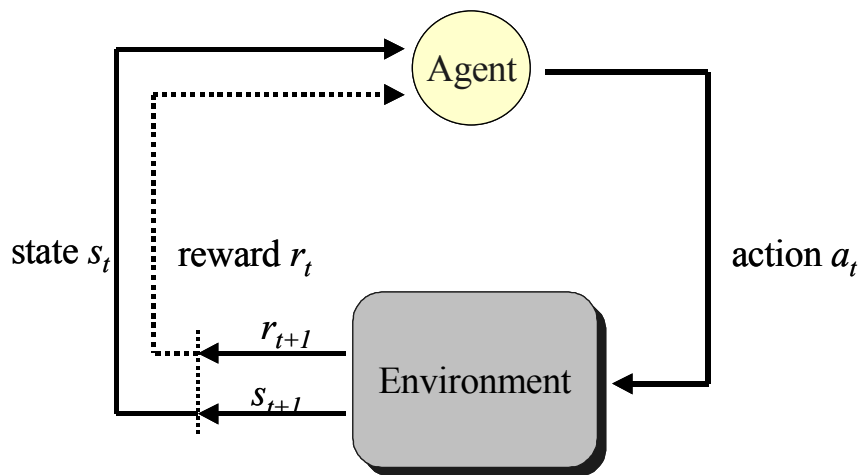


그림 1: 학습자와 환경간의 상호작용

4.1.3 보상 함수(Reward Function)

보상 함수는 학습자의 행동에 대한 환경으로부터의 반응으로 보통 스칼라 (scalar)값으로 주어진다. 강화 학습에 주어지는 보상은 감독 학습에서처럼 지시적인(instructive) 특성을 갖는 것이 아니라 평가적인(evaluative) 특성을 갖고 있으며 아래 식으로 표현한다.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (13)$$

위 식에서 γ 는 할인 상수(discount factor)로 미래에 받게 될 보상이 현재의 상태의 가치나 상태-행동의 가치에 반영되는 정도를 조절한다. 즉 γ 의 값이 1에 가까울수록 t 시각 이후에 받게 될 보상을 할인하지 않고 반영하게 된다.

4.1.4 가치 함수(Value Function)

가치함수는 학습이 계속되는 동안 누적되는 보상과 현재의 상태에 대한 가치를 평가하는 함수이다. 일반적인 가치함수는 아래 식과 같이 표현한다.

$$V(s) = V(s) + \alpha [V(s') - V(s)] \quad (14)$$

가치함수는 현재 상태의 가치를 평가하여 다음 상태의 가치를 추정하여 학습하는 규칙이다. 여기에서 α 는 현재 상태의 변화량 즉, 학습률을 조절하는 상수이다. 학습 목표에 가깝게 추정된 값 $V(s_{t+1})$ 와 현재 상태의 값의 차이를 α 만큼씩 현재의 상태를 변화시켜 가는 형식으로 학습 목표를 달성한다. 가치 함수

의 이러한 특성을 Bootstrap이라 하는데 Bootstrap 특성이란 목적에 가깝다고 추정된 값과 현재의 값의 차를 계속적으로 줄여가면서 목적에 다가가는 성질을 말한다. 구체적으로는 상태에 대한 가치함수 아래 식 또는 상태-행동 쌍에 대한 가치함수 식으로 나뉘어지는데 이는 t+1 이후에 시간에 대한 기대값으로 표현된다.

$$V^\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right] \quad (15)$$

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right] \quad (16)$$

π 는 학습자의 행동 선택에 기준이 되는 정책을 말하며 각 상태의 가치 또는 어떤 상태에서 어떤 행동을 선택했을 때의 가치를 계산하는 것을 무한히 반복하게 되면 결국 최종 목적에 수렴하게 된다.

4.2 강화 학습을 이용한 정보 검색

이 장에서는 위에서 설명한 기본적인 강화 학습을 웹 정보를 검색하는 문제에 적용하면 다음과 같다. 우선, 여러 치즈조각이 놓여져 있는 미로를 찾아가는 쥐를 생각해 보면 쉽게 이해가 된다. 주어진 미로의 한 위치에서 쥐는 치즈를 찾기 위해서 다음 위치로 움직이는 행동을 하고 치즈가 있는 위치를 찾아내면 이에 대한 보상으로 치즈를 먹을 수가 있다. 쥐가 우연히 치즈가 있는 위치로 한번에 가게 되면 즉각적인 보상을 받지만 결국에는 여러 개의 치즈를 최단 시간 안에 다 먹기 위해서는 미래의 보상을 고려하면서 최적의 행동을 선택해야한다.

이와 유사하게 웹 상에서 사용자가 원하는 웹 문서를 찾는 문제도 생각해 볼 수 있다. 한 웹 페이지가 주어져 있고 이 페이지에 여러 하이퍼링크가 있을 때, 에이전트는 사용자가 관심 있어 하는 문서를 찾기 위하여 관련 링크를 따라 가면서 정보를 검색하게 된다. 이때 각 웹 페이지들은 상태(state)로 정의하고, 하이퍼링크를 따라가는 것을 행동(action)으로 정의할 수 있다. 각 상태 s 에서 에이전트는 그에 따른 보상값(reward) $R(s)$ 를 받고 이 상태에서 하이퍼링크를 따라가는 행동(action)을 취하면 이 행동이 에이전트의 최종 목적(찾고자 하는 웹페이지)에 있어서 얼마나 유용한가를 가치함수 $Q(s,a)$ 를 통해서 평가할 수 있다. 이 웹 탐색의 목적은 사용자가 원하는 정보가 있는 웹 페이지를 찾는 것이 된다. 가치함수 $Q(s,a)$ 는 한 웹 페이지에서 사용자가 관심도 나타내는 하이퍼링크를 따라가는 행동에 대한 할인된(discounted) 보상값의 합이 된다. 다시 말하면 웹 페이지를 나타내는 단어들과, 하이퍼링크의 앵커 텍스트상의 단어들을 tf-idf 벡터 표현으로 나타내어서 이를 사용자 프로파일상의 단어들과의 유사도를 조사하고 이를 보상값으로 정의하여 가치함수 $Q(s,a)$ 값을

구했다. 즉 웹 페이지들을 전처리 과정을 통하여 tf-idf 벡터로 표현하고 이 웹 페이지들의 하이퍼링크에 나오는 단어들과 이 하이퍼링크를 따라갔을 때 나오는 페이지의 타이틀이나 헤더의 단어를 이용하여 보상함수 $R(s)$ 을 구하고 이를 다음의 식을 통하여 Q 값을 학습하였다.

$$p = (w_{1,p}, w_{2,p}, \dots, w_{t,p}) \text{ 사용자 프로파일 벡터} \quad (17)$$

$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j}) \text{ 문서나 링크에 대한 벡터} \quad (18)$$

$$\cos(d_j, p) = \frac{d_j \cdot p}{|d_j| \cdot |p|} = \frac{\sum_{i=1}^t w_{i,j} \cdot w_{i,p}}{\sqrt{\sum_{i=1}^t (w_{i,j})^2} \cdot \sqrt{\sum_{i=1}^t (w_{i,p})^2}} \quad (19)$$

$$Q(s_t, a) = \sum_{k=0}^{\infty} \gamma^k R(s_{t+1+k}) \quad (20)$$

학습된 결과를 가지고 에이전트가 웹 페이지의 하이퍼링크를 따라갈 때 사용자의 관심도에 나타난 단어들의 Q 값이 가장 큰 하이퍼링크를 따라 가게 됨으로써 너비 우선(breadth-first search)방식의 검색 보다 더 효율적인 검색을 할 수 있다. 이는 어느 정도 검색할 영역이 한정되어 있는 사항에서 사용자의 프로파일상의 변화가 없다는 가정 하에서는 성능에 있어서 많은 향상을 가져올 수 있다. 그러나 실제로 사용자의 프로파일상의 변화는 어느 정도 관심도 변화에 따라서 달라질 수 있다. 본 논문에서 제안한 방법을 정리하면 다음과 같다.

0. 사용자의 관심도를 설정한다. 사용자 프로파일 벡터 생성
1. 주어진 웹 페이지를 스캔한다. 문서 벡터 생성 과정
2. 하이퍼링크를 조사한다.
3. 링크의 앵커 텍스트와 관련 페이지의 타이틀, 헤더정보를 조사한다. 링크 벡터 생성 과정
4. 링크들의 유사도를 계산하여 유사도가 제일 큰 링크를 선택한다. 보상값 Q 계산
5. 단계 1로 가서 반복한다.
6. 만족된 결과가 있으면 리턴 한다.

표 2: 정보 검색을 위한 의사코드

일반적으로 강화학습을 이용한 정보검색은 많은 하이퍼링크 중에서 보상값이 높은 링크를 선택하는 경향을 보이게 된다. 이에 따라 많은 학습을 시킨 후에 나오는 결과는 자신에게 들어오는 많은 링크들을 갖고 있는 페이지인 전거나 자신에서 나가는 많은 링크들을 갖고 있는 허브 사이트를 찾게 되는 장점이 있다. 이러한 과정을 아래 그림 2를 통하여 알 수 있다.

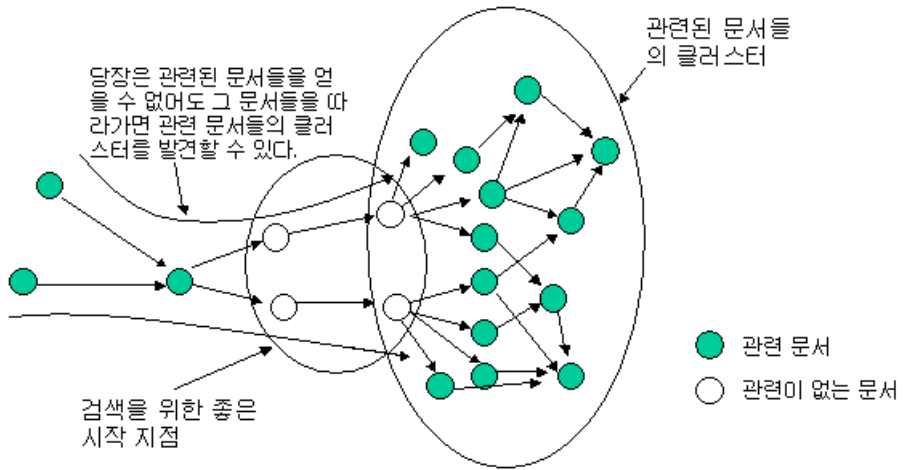


그림 2: 에이전트의 탐색 모습

5. 실험 및 결과

5.1 실험 데이터 및 방법

첫 번째 실험에 사용된 데이터는 학회 논문제출을 요청하는 홈페이지(call-for-paper)와 학회 안내 홈페이지(conference homepage)를 대상으로 총 200개의 문서를 수집하였고, 이에 관련된 하이퍼링크는 약 850개를 만들었다.

이 문서들의 내용은 인공지능에 관련된 해외 학회에 대한 정보와 논문 제출 안내를 담고 있다. 수집된 웹 문서에 존재하는 하이퍼링크를 살펴보면 파일을 가지고 있는 곳을 연결해주는 하이퍼링크와 끊어진 링크와 시스템 관리를 처리하는데 사용되는 링크들이 존재하는데 정확한 실험을 위해서 이를 제외하였다. 실험조건상의 사용자 프로파일 색인어 수는 10개로 하였고, 사용자 관심도의 변화는 없는 것으로 가정하였다.

에이전트 관련 학회를 검색의 목적으로 하였고 에이전트 관련 학회를 찾기 위해 따라간 하이퍼링크의 퍼센트와 발견한 에이전트 관련 학회 문서의 퍼센트로 하여 비교하였다. 비교대상으로는 일반적인 성능비교를 하는 너비 우선 검색(BFS)으로 하였다.

또 다른 실험에서는 카네기 멜론(CMU)대학의 텍스트 학습 그룹에서 제공하는 WebKB 데이터를 사용하였다. 이 데이터 집합은 여러 대학의 컴퓨터 공학과 홈페이지에서 수집된 웹 페이지들이다. 전체 1051개의 웹 페이지가 있으며, 230개의 전공교과에 관련된 웹 페이지와 821개의 그 외의 것으로 분류되어 구성되어 있다. 또한 전문(fulltext)이 나와 있는 데이터와 그 페이지를 가리키

고 있는 하이퍼링크상의 앵커 텍스트만을 가지고 있는 데이터로 분류가 되어
 져 있다. 이 중에서 본 논문에서는 821개의 웹 페이지로 구성된 데이터 집합으
 로 실험을 하였다. 이 데이터들은 주로 교수진을 소개하는 웹 페이지와 그와
 관련된 연구와 학회 및 논문에 관한 내용들이 담겨져 있다. 첫 번째 실험과 마
 찬가지로 따라간 하이퍼링크의 퍼센트에 대해서 사용자가 찾고자 하는 문서의
 퍼센트로 비교를 하였다. 역시 비교대상으로 너비 우선 방식으로 하였다. 다른
 실험 방법으로는 200개와 500개의 하이퍼링크를 따라갔을 때까지 사용자가 원
 하는 페이지들이 검색된 문서 수를 조사하였다.

	웹 페이지 수	하이퍼링크 수	페이지의 내용
실험1 데이터	200개	850개	인공지능학회 관련 웹 페이지
실험2 데이터	821개	7천 여개	컴퓨터 공학과 홈페이지에서 수집된 웹 페이지

표 3: 실험 데이터에 대한 설명

5.2 실험 결과 및 분석

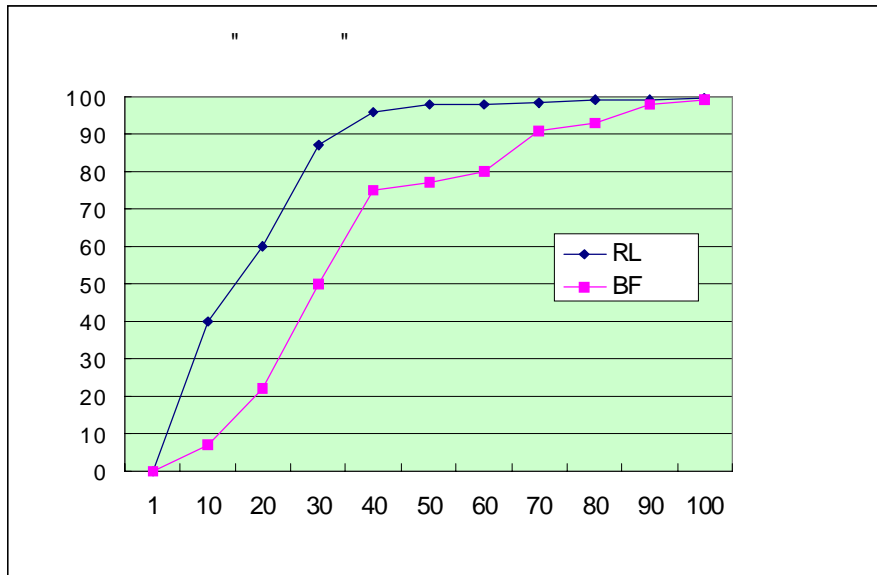


그림 3 : 실험1의 결과

실험 1의 결과를 살펴보면 RL(강화 학습)과 BF(너비 우선)을 적용하여 비교하였는데 x축은 문서를 찾기 위해서 에이전트가 따라간 하이퍼링크의 퍼센트 비율이고 y축은 에이전트가 발견한 문서에 대한 퍼센트를 나타낸다. 그림 3을 보면 알 수 있듯이 강화학습을 이용한 RL이 너비 우선 검색인 BF보다 거의 3배에 가까운 성능 효율을 보이고 있다. 최종적으로 많은 하이퍼링크를 따라가면 RL이나 BF가 원하는 문서를 거의 다 찾아 주기는 한다. 하지만 초기 분포(하이퍼링크의 30%이내)에서는 강화학습을 이용한 검색이 적은 하이퍼링크를 가지고 더 정확하게 원하는 정보를 찾아주는 것을 알 수 있다. 이는 학습 통하여 시작점에서 많지 않은 링크를 가지고도 사용자의 관심도를 반영하여 검색을 하기 때문이다. 그림 4은 그림 3의 초기 분포를 자세히 보이고 있다. 다른

검색 조건에서도 이와 유사한 분포를 보인다.

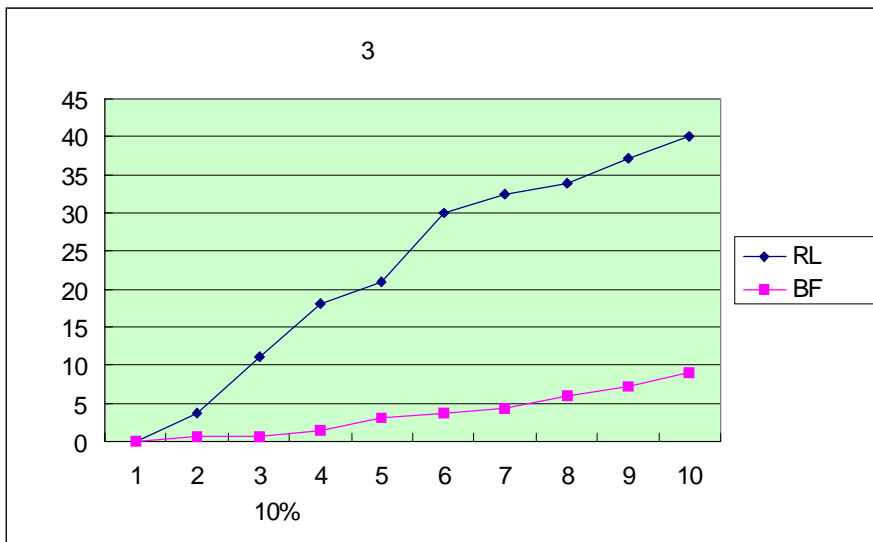


그림 4: 그림 3의 초기 분포 분석

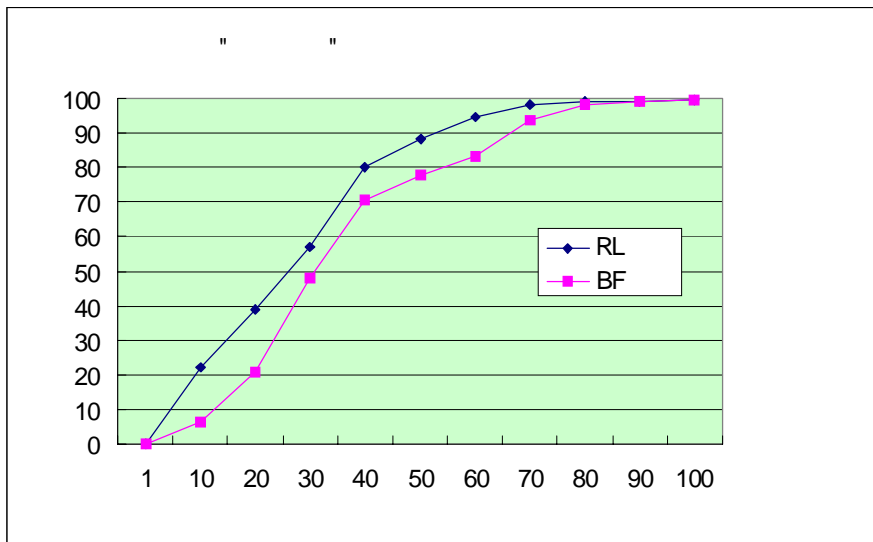


그림 5 : 실험 2의 결과

실험 2의 결과를 살펴보면 실험 1의 경우와는 약간 다르게 나타나고 있다. 전체적으로 보았을 때는 BF검색이 실험 1의 경우보다는 더 향상된 것 같다. 그러나 초기 30%분포를 비교해 볼 때 약 2배정도 RL검색이 더 좋은 성능을 보이고 있다. 이러한 이유를 분석해 보면 데이터의 내용과 연관이 있는 듯 보인다. 실험 2의 데이터는 컴퓨터공학과 관련 페이지이므로 사용자 프로파일상의 어휘와 관련 없는 내용들이 많이 분포하고 있을 수 있다. 즉, 특정 주제와 관련이 적게 되면 검색 효율성도 떨어질 수 있음을 나타내준다고 할 수 있다. 이는 검색하는 도메인을 줄이면 해결될 수 있는 문제로 생각되어진다.

다음은 따라간 하이퍼링크의 개수를 고정시키고 실험을 한 경우에 대한 결과를 보이고 있다. 그림 6과 7에서 볼 수 있듯이 처음 200개까지의 경우에는 RL검색이 BF검색에 비하여 훨씬 많은 관련문서를 찾고 있지만, 500개까지의 경우에는 거의 비슷한 성능을 보인다. 이는 BF검색이 어느 정도의 링크를 따라가다 보면 일정이상의 성능을 보임을 알 수 있다.

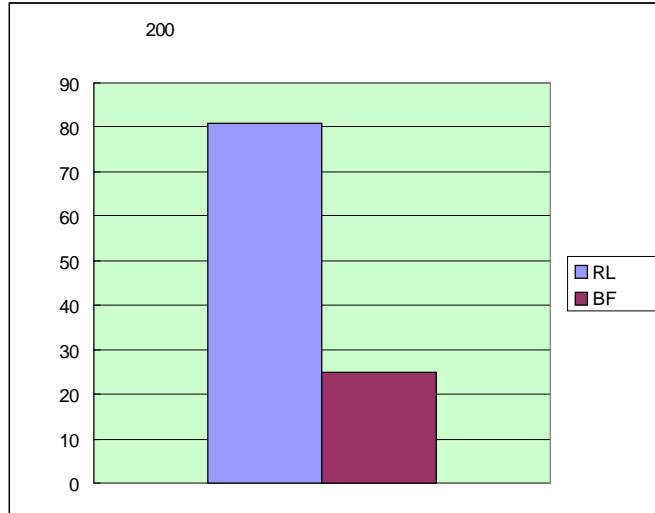


그림 6: 따라간 링크 개수를 200개로 제한할 경우의 실험 결과

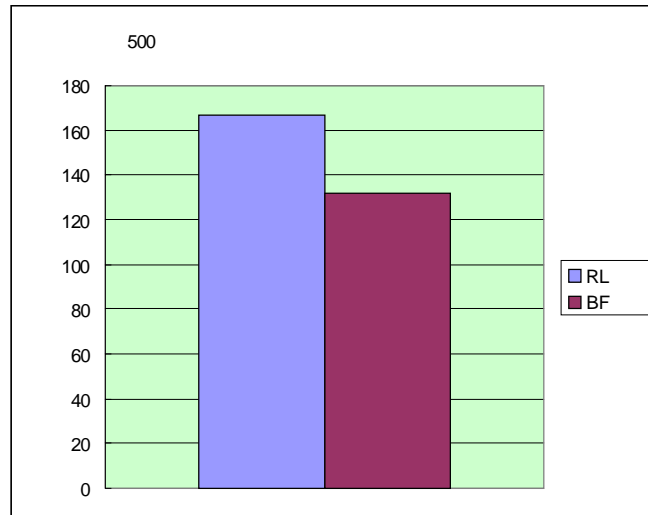


그림 7: 따라간 링크 개수를 500개로 제한할 경우의 실험 결과

6. 결론 및 향후과제

지금까지 살펴본 바에 따르면 범용 검색엔진에서 사용하고 있는 검색 알고리즘인 너비 우선 방식의 검색보다 웹 문서의 구조적인 특징인 하이퍼 링크 구조를 대한 분석을 통한 강화학습 알고리즘을 이용하여 사용자의 관심도에 따른 검색방식이 더 좋은 성능향상을 보였다. 본 논문에서 제안하고 있는 검색 방법이 모든 영역에 대해서 검색하는 문제에 있어서는 효율성이 검증되고 있지는 않다. 하지만 본 실험에서 제시된 결과에 따르면 사용자의 관심도에 관련된 영역을 검색하는 문제에 있어서는 성능이 향상 될 수 있음을 보여 주었다. 이는 찾고자 하는 영역을 줄이는 대신, 보다 더 정확하고 빠른 검색을 필요로 하는 특정 주제어 관련 검색엔진에 있어서 크게 도움을 줄 수 있다. 앞으로의 연구에 있어서는 본 연구에서 제시한 방법과 사용자의 관심도 변화에 따른 검색 방법을 연계하는 연구가 진행되어야 할 것이다. 이번 논문에서 제시된 방법을 바탕으로 온라인 상의 하나의 시스템에서 새로운 웹 정보 문서가 새로 생겨났을 때 이 문서에 대한 강화학습을 이용한 효율적인 검색을 통하여 사용자에게 추천해 주고 이를 통하여 사용자의 관심도를 학습하는데 중점을 두고 검색된 문서를 여과하는 방법을 생각해 볼 수 있다.

참고 문헌

- [Brin and Page, 1998] Brin, S. and Page, L., The Anatomy of a Large-scale Hypertextual Web Search Engine, *The Seventh International World Wide Web Conference (WWW7)*, pp. 107-117, 1998.
- [Boyan et al., 1996] Boyan, J., Freitag, D., and Joachims, T., A Machine Learning Architecture for Optimizing Web Search Engines, *Proceedings of the AAAI Workshop on Internet-Based Information Systems*, pp. 1-8, 1996.
- [Callan et al., 1992] Callan, J. P., Croft, W. B. and Harding, S. M., The INQUERY Retrieval System, *Proceedings of the Third International Conference on Database and Expert Systems Applications*, Springer, pp. 78-83, 1992.
- [Carriere and Kazman, 1997] J. Carriere and R. Kazman. WebQuery: Searching and visualizing the Web through connectivity. In *6th Int. WWW Conf.*, USA, 1997.
- [Chakrabarti et al., 1997] Chakrabarti, S., Dom, B., Gibson, D., Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tomkins, A., Experiments in Topic Distillation, *ACM-SIGIR '98 Post-Conference Workshop on Hypertext Information Retrieval for the Web*, 1998.
- [Cho et al., 1998] J. Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Computer Networks and ISDN System*, volume 30, 1998.
- [Google, 2000] Google Inc. Press release: "Google launches world's largest

search engine." June 26, 2000.

- [Joachims et al., 1997] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A tour guide for the WWW. In *Proceedings of IJCAI-97*.
- [Kleinberg, 1998] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pp. 668-677, 1998.
- [Li, 1998] Y. Li. Toward a qualitative search engine. *IEEE Internet Computing*, 2(4):24-29, 1998.
- [Lyman et al., 2000] P. Lyman, H. Varian, J. Dunn, A. Strygin, and K. Swearingen. How much information? School of Information Management and Systems, Univ. of California at Berkeley, 2000.
- [Maron and Kuhns, 1960] Maron, M. E. and Kuhns, J. L., On Relevance, Probabilistic Indexing and Information Retrieval, *Association for Computing Machinery*, 7(3), pp. 216-244, 1960.
- [McCallum et al., 1999] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Building domain-specific search engines with machine learning techniques. In *AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace*, 1999.
- [Mitchell, 1997] Tom M. Mitchell, *Machine Learning*, McGraw-Hill Com. Inc., 1997.
- [Robertson and Sparck, 1976] Robertson, S. E. and Sparck Jones, K., Relevance Weighting of Search Terms, *Journal of the American Society for Information Science*, 27, pp. 129-146, 1976.
- [Salton, 1989] Salton, G., *Automatic Text Processing*, Addison-Wesley, pp. 279-281, 1989.

- [Salton et al., 1983] Salton, G., Fox, E. A., and Wu, H., Extended Boolean Information Retrieval, *Communications of the ACM*, Vol. 26, No. 11, pp. 1022-1036, 1983.
- [Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [Turtle and Croft, 1991] Turtle, H. and Croft, W. B., Evaluation of an Inference Network-based Retrieval Model, *ACM Transactions on Information Systems*, Vol. 9, No. 3, pp. 187-222, 1991.
- [Yuwono and Lee, 1996] B. Yuwono and D. L. Lee. Search and ranking algorithms for locating resources on WWW. In *Proc. of the Int. Conference on Data Engineering*, pp. 164-171, 1996.

Abstract

This paper presents the method for searching web pages on a particular kind or on a particular topic with increased accuracy using reinforcement learning, a branch of machine learning.

General purpose search engines traverse web link graphs using crawlers which collect the informations accessing the information sources in the WWW, this crawlers use breadth-first search strategy. But this breadth-first search strategy is not efficient in using the information on the hyperlinks which are the structural feature of the web pages which are written in HTML(Hypertext Markup Language).

Concerning the problem of following hyperlinks in the WWW for finding specific web page, reinforcement learning provides a formalism for measuring the utility of actions that give benefit in the future. In relation to using hyperlink's informations, there have been several studies of ordering algorithms which analyze the hyperlinks on the web pages.

Experiments have been performed on an artificial dataset and WebKB dataset. The first dataset consists of 200 web pages, 800 hyperlinks about artificial intelligent conference web pages, second dataset consists of about 1000 web page, 7000 hyperlinks. Our empirical results show that the proposed method performs more efficient search, requiring exploration of fewer hyperlinks than the breadth-first search method does.

Keyword: Information retrieval, hyperlink, reinforcement learning, breadth-first search