

다중 에이전트 환경에서 효율적인 강화학습 시스템

홍정환^o 강진범 최중민

한양대학교 컴퓨터공학과 지능시스템 연구실

{jhong^o, jbkang, jmchoi}@cse.hanyang.ac.kr

Efficient Reinforcement Learning System in Multi-Agent Environment

Junghwan Hong^o JinBeom Kang Joongmin Choi

Dept. of Computer Science and Engineering, Hanyang University

요 약

강화학습은 환경과 상호작용하는 과정을 통하여 목표를 이루기 위한 전략을 학습하는 방법으로써 에이전트의 학습방법으로 많이 사용한다. 독립적인 에이전트가 아닌 상호 의사소통이 가능한 다중 에이전트 환경에서 에이전트의 학습정보를 서로 검색 및 공유가 가능하다면 환경이 거대하더라도 기존의 강화학습보다 빠르게 학습이 이루어질 것이다. 하지만 아직 다중 에이전트 환경에서 학습 방법에 대한 연구가 미흡하여 학습정보의 검색과 공유에 대해 다양한 방법들이 요구되고 있다. 본 논문에서는 대상 에이전트 학습 정보와 주변 에이전트들의 학습 정보 사이에 편집거리를 비교하여 유사한 에이전트를 찾고 그 에이전트 정보를 강화학습 사전정보로 사용함으로써 학습속도를 향상시킨 ED+Q-Learning 시스템을 제안한다.

Key words : 다중 에이전트, 강화학습, Q-Learning, 편집거리

1. 서 론

최근 몇 년 사이에 인공지능 분야에서 사용자를 대신하여 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어라고 할 수 있는 '에이전트(agent)'라는 개념이 등장하면서 기존의 연구 범위에서 벗어나 넓은 범위에 걸쳐 새롭게 적용될 수 있는 가능성을 인식시키기 시작했다.[2].

또한 에이전트가 동시에 다수가 존재하는 다중 에이전트 환경으로도 발전되었다. 다중 에이전트 환경은 하나의 에이전트가 해결하지 못하는 복잡한 작업을 해결하기 위해 여러 에이전트들로 구성되는 환경이다. 이러한 환경에서 에이전트들은 서로 협동을 통해 작업을 할 수도 있고 서로 경쟁을 통해 작업을 할 수도 있다.[3].

에이전트는 능동적인 목표 달성을 하기 위해 환경에 대한 지식을 습득하는 행위인 학습(learning)을 하게 되는데 강화학습(reinforcement learning)은 에이전트의 학습 방법들 중에서 비교적 많이 사용되고 있는 방법이다.

강화학습은 동적인 환경에서 행동의 시행착오를 거쳐 환경으로부터 주어지는 보상(reward)을 최대화하기 위한 학습 방법으로 행동을 미리 예측할 수 있고, 변화하는 환경에서도 학습할 수 있다.

강화학습의 대표적 알고리즘인 Q-Learning은 상태-행동의 가치 함수인 Q-함수를 계산하고 가치 함수를 사용해 한 단계를 미리 예측하고 그 결과를 최적의 전략으로 이끈다. 또한 모든 상태-행동 쌍을 하나의 룩업 테이블에 저장한다. 그러나 기존의 Q-Learning은 몇 가지

문제점을 가지고 있다.[1].

첫째, 새로운 환경에서 학습을 할 때 Q-Learning 알고리즘 특징상 룩업 테이블이 초기화되기 때문에 특정한 횟수 이상의 학습을 하기 전까지 학습정보는 사용이 무의미해진다. 둘째, 학습 환경이 거대하거나 복잡한 경우 상태-행동 쌍에 대한 경우의 수가 너무 많아져서 학습속도가 느려지며 모든 상태-행동 쌍을 경험하기 어렵다.

과거에는 Q-Learning의 문제점을 에이전트가 독립적으로 존재하는 환경에서 해결하려는 연구가 주를 이루었다. 그러나 최근 다중 에이전트 환경이라 할 수 있는 유비쿼터스(ubiquitous) 환경이 등장함으로써 Q-Learning의 문제점을 해결하기 위해 단일 환경 및 다중 환경에서 에이전트들 간에 의사소통을 통해 학습된 정보를 검색하고 그 정보를 공유하는 다양한 방법들이 요구되고 있다.

본 논문에서는 다중 환경에 에이전트들이 존재할 때 Q-Learning의 문제점을 해결하기 위한 하나의 시스템으로 여러 에이전트 중 유사한 환경에 존재하는 에이전트의 정보를 활용하기 위해 편집거리를 사용하여 유사한 에이전트를 검색하고 그 정보를 Q-Learning에 활용함으로써 학습속도 향상에 중점을 둔 ED+Q-Learning 시스템을 제안한다.

2. 관련연구

새로운 환경에서 Q-Learning을 하는 에이전트는 학습 초기에 초기화된 Q함수와 적은 경험을 바탕으로 행동선택을 해야 하므로 학습속도가 느려진다는 문제점을 가지고 있다. 이러한 문제점은 새로운 환경에 대한 사전정보

를 얻을 수 있다면 해결될 수 있다. 최근 다중 에이전트 환경이 등장하면서 다른 에이전트가 학습한 정보를 사전 학습 정보로 활용하는 연구가 다양하게 이루어지고 있다.

2.1 Episodic Q-Learning

Q-Learning에서 임의의 초기 상태를 선택하고 행동을 취하면서 목표에 도달할 때까지의 과정을 하나의 에피소드(episode)라고 말한다. Episodic Q-Learning은 이러한 에피소드를 여러 상태와 행동 쌍들의 순차적 집합이라고 보고 [식 1]과 같이 에피소드의 모든 상태와 행동 쌍 순차의 역순으로 Q 테이블의 Q 값을 갱신하는 학습법이다.[4].

$$Q(S_k, a_k) \leftarrow (1 - \alpha)Q(S_k, a_k) + \alpha(r_k + \gamma \max_a Q(S_{k+1}, a)) \quad [\text{식 1}]$$

$$k = t, t-1, t-2, \dots, 0$$

Episodic Q-Learning은 완료된 에피소드를 사용해 Q 값을 갱신하는 학습법이기에 때문에 여러 에이전트들이 에피소드를 공유할 수 있는 다중 에이전트 환경에 적합한 학습법이다. 또한 Episodic Q-Learning을 사용함으로써 기존의 Q-Learning에 비해 Q 테이블이 신속하게 채워지므로 학습속도가 빨라지게 되었다. 그러나 서로 다른 환경에 위치한 에이전트들 간에 에피소드를 공유하고 학습을 할 경우 불필요한 에피소드가 학습될 가능성이 문제점으로 존재한다.

2.2 Multi-Agent MAXQ 알고리즘

MAXQ는 계층적 강화학습[5]의 한 종류로써 목표를 이루기 위한 여러 작업들에 대해서 작업 계층을 만들고 현재 상태 s 에서 작업 i 에 대한 $Q_i(i, s, a)$ 값은 하위 작업 a 에 대한 기대누적보상 값 $C_i(i, s, a)$ 와 $V_i(i, s)$ 값의 합으로 구하게 된다. MAXQ는 단일 에이전트의 학습에 중점을 두었기 때문에 다중 에이전트 환경에 적합하지 않다. Multi-Agent MAXQ 알고리즘은 기존의 단점을 해결하기 위해서 MAXQ를 확장하여 [식 2]와 같이 다른 에이전트들의 작업($a^1 \dots a^j \dots a^n$)을 동시에 고려하여 $Q_i^j(i, s, a^1, \dots, a^j \dots a^n)$ 를 구하는 방법이다. 다른 에이전트의 작업을 고려하기 때문에 에이전트들 간에 중복되는 작업이 없이 학습이 가능하므로 빠른 학습이 가능하다.[6].

$$V_i^j(i, s, a^1, \dots, a^{j-1}, a^{j+1}, \dots, a^n) = \begin{cases} \max_{a_j} Q_i^j(i, s, a^1 \dots a^j \dots a^n) & \text{if composite}(i) \\ \sum_s P(s'|s, i)R(s'|s, i) & \text{if primitive}(i) \end{cases}$$

$$Q_i^j(i, s, a^1 \dots a^j \dots a^n) = V_i^j(a^j, s) + C_i^j(i, s, a^1 \dots a^j \dots a^n) \quad [\text{식 2}]$$

2.3 WSS(Weighted Strategy Sharing)

WSS는 다중 에이전트 환경에서 나중에 좀 더 유용한 에이전트들의 학습정보를 사용하기 위해서 각 에이전트가 받는 보상에 따라서 숙련도를 부여하는 학습법이다.[7].

```

(1) Initialize
(2) while not EndOfLearning do
(3) begin
(4) If IndividualLearningMode then
(5) begin { Individual Learning}
(6) Q-Learning
(7)  $e_i := \text{UpdateExpertness}(r_i)$ 
(8) end
(9) else { Cooperative Learning}
(10)begin
(11) for  $j=1$  to  $n$  do
(12)  $e_j := \text{GetExpertness}(A_j)$ 
(13)  $Q^{new} := 0$ 
(14) for  $j=1$  to  $n$  do
(15)begin
(16)  $W_{ij} := \text{ComputeWeights}(i, j, e_i, \dots, e_n)$ 
(17)  $Q_j^{old} := \text{Get}Q(A_j)$ 
(18)  $Q^{new} := Q^{new} + W_{ij} * Q_j^{old}$ 
(19)end
(20)end
(21)end
    
```

[그림 1] WSS 알고리즘

WSS 알고리즘은 [그림 1]과 같이 Individual Learning 과 Cooperative Learning 2가지 방식으로 이루어진다. 에이전트들은 초기에 Individual Learning 방식으로 학습을 시작하며 특정횟수 이상을 초과하면 Cooperative Learning 방식으로 전환하여 학습을 하게 된다. Individual Learning 방식은 기존의 Q-Learning 방식과 동일하며 Cooperative Learning 방식은 좀 더 숙련도가 높은 에이전트의 Q 테이블의 Q 값과 자신의 숙련도와 다른 에이전트의 숙련도 차이를 가중치로 두어 가중치를 Q 값에 곱해줌으로써 자신의 Q 테이블에 반영하게 된다. 자신보다 좀 더 유용한 에이전트의 정보를 이용함으로써 빨리 목적지에 도달할 수 있다.

3. ED+Q-Learning 시스템

3.1 다중 환경 에이전트에 적합한 Q-Learning

다중 에이전트 환경에서는 에이전트들 간의 중개자 역할을 하는 facilitator를 통해서 주변에 어떠한 에이전트들이 존재하는지 알 수 있으며 존재하는 에이전트들과 의사소통도 가능하다. 이러한 facilitator를 통해 자신과 유사한 환경에 위치한 에이전트를 찾아 미리 학습한 정보를 이용할 수 있다면 Q-Learning 문제점인 모든 상태-행동 쌍을 경험하지 않고도 적절한 행동 예측이 가능하고 Q 테이블이 초기화 되거나 많은 상태-행동이 있더라도 빠른 학습이 가능하다.

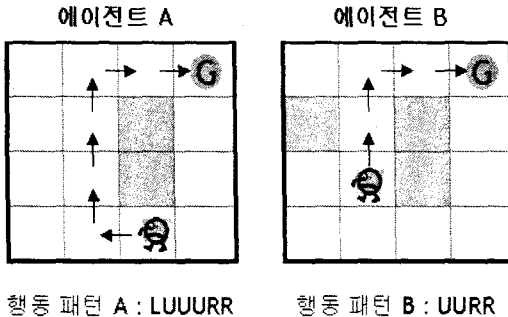
기존에는 [4][6][7]과 같이 단일 환경에 존재하는 에

이전트들 간에 facilitator를 통해 서로 정보를 공유함으로써 Q-Learning의 문제점을 해결하는 학습법이 연구되었다. 그러나 이러한 학습법도 서로 다른 환경에 에이전트들이 존재한다면 에이전트들 간에 학습정보가 다르기 때문에 적용시키기 어렵다.

본 논문에서는 기존에 두 개의 문자열을 비교할 때 사용했던 편집거리를 사용하여 다중 환경에 존재하는 에이전트들 중 자신과 유사한 환경에 위치한 에이전트를 찾고 유사한 에이전트의 학습정보를 활용함으로써 학습속도 향상을 고려한 시스템을 제안하고 있다.

3.2 편집거리를 사용한 에이전트 유사도 판단

다중 환경 에이전트들 간에 편집거리를 활용하기 위해서 각 에이전트의 행동을 편집거리 연산이 가능한 형태로 바꿔줘야 된다. 예를 들어 목적지를 찾아가는 에이전트의 경우 이동경로를 다음의 [그림 2]에서처럼 상(Up)하(Down)좌(Left)우(Right)와 같은 문자열로 치환한다. 치환된 행동을 목적 달성할 때까지 저장하게 되면 하나의 행동패턴 A가 생성되는데 행동패턴 A와 다른 에이전트의 행동패턴 B로 편집거리 연산을 시작한다. 편집거리 연산을 시작하기 전에 행동패턴 A를 행으로, 행동패턴 B를 열로 두고 연산을 시작한다. Facilitator에 등록된 모든 에이전트들의 행동패턴과 연산을 하여 적은 연산수가 나온 에이전트를 가장 유사한 에이전트라고 본다. 예를 들어, [그림 2]와 같은 행동패턴이 있을 경우 연산결과 는 [표 1]과 같이 나타난다.



[그림 2] 행동치환 / 행동패턴 생성

그러나 편집거리 연산수가 적게 나온 에이전트라고 해도 전체가 아닌 특정수의 에이전트들 중에 낮은 연산수를 가진 에이전트이기 때문에 절대적으로 유사한 에이전트라고 장담할 수 없다. 그러므로 본 논문에서는 추가적으로 편집거리 연산수에 대해서 특정 임계치를 적용함으로써 절대적으로 유사한 에이전트를 선별하게 된다. 이때 임계치는 실험을 통한 수치로 정해진다.

3.3 ED+Q-Learning 시스템 구조

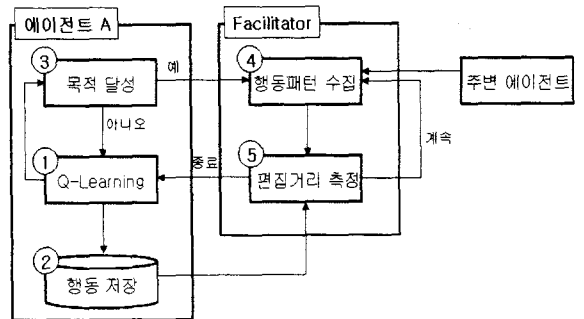
다중 환경 에이전트들 간에 편집거리를 사용하여 유사

[표 1] 행동패턴 A와 B의 편집거리 계산

		U	U	R	R
	0	1	2	3	4
L	1	1	2	3	4
U	2	1	1	2	3
U	3	2	1	2	3
U	4	3	2	2	3
R	5	4	3	2	2
R	6	5	4	3	2

한 에이전트를 판단한 후 유사한 에이전트의 정보를 이용하여 학습을 하는 전체적인 시스템 구조는 [그림 3]과 같다.

시스템에서 에이전트 A는 학습을 하는 주체로서 크게 자신이 학습을 하는 Q-Learning 단계와 행한 행동을 저장하기 위한 행동 저장 단계로 나누며 학습 중 목적을 달성하게 되면 자신과 유사한 에이전트를 찾기 위해 facilitator에게 지금까지 학습한 행동패턴 정보를 넘긴다. 이러한 에이전트 A의 행동패턴 정보를 받은 facilitator는 자신에게 등록되어 있는 주변 에이전트들의 행동패턴과 각각 편집거리 연산으로 유사도를 측정하여 1차적으로 유사한 에이전트들을 선별한 후 실험을 통해 얻은 특정 임계치를 사용해서 선별된 에이전트들에 대한 객관적 유사도를 판단함으로써 가장 유사한 에이전트를 선별해 낸다. 다음으로 유사한 에이전트가 학습했던 Q 테이블 정보를 요청한 에이전트 A에게 전송해 주고 정보를 받은 에이전트 A는 자신의 Q 테이블에 반영을 한 후 다시 학습을 시작한다.



[그림 3] 시스템 구조

이와 같이 ED+Q-Learning 시스템은 다중 에이전트 환경의 장점인 의사소통으로 다른 에이전트의 학습정보와 자신의 편집거리 비교하여 찾아낸 가장 유사한 에이전트의 정보를 학습에 활용함으로써 기존의 Q-Learning의 문제점을 해결하였다.

4. 실험 및 결과

본 절에서는 제안한 ED+Q-Learning 시스템의 타당성

을 검증하기 위해, 강화학습으로 학습한 에이전트의 성능을 평가하는데 많이 사용되는 격자 형태로 구성된 공간에서 목적지를 찾는 방법을 사용했다.

4.1 실험방법

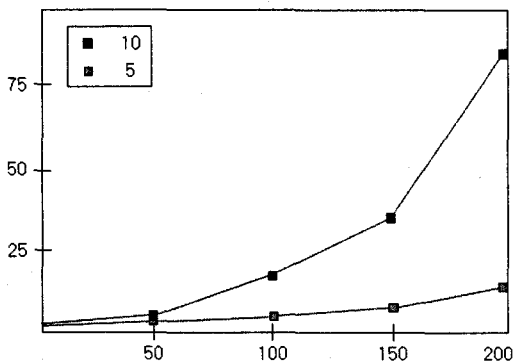
본 논문에서는 ED+Q-Learning 시스템의 타당성을 증명하기 위해 서로 다른 조건이지만 같은 크기의 환경에 에이전트를 위치시키고 각각 특정 횟수의 학습을 수행한다. 학습을 수행한 후 학습정보를 학습하지 않은 에이전트와 ED+Q-Learning을 함으로써 학습을 하지 않은 에이전트가 학습정보를 통해 자신과 유사한 환경을 찾아내는가에 대해서 실험을 하였다.

4.2 실험결과

ED+Q-Learning이 유사한 환경을 잘 찾아내는가를 알아보기 위해서 본 실험에서는 학습 횟수에 따른 분포를 [표 2] 및 [그림 5]로 나타내었다.

[표 2] 유사 환경인지 실험결과 1

	5-100	5-200	5-250	10-100	10-200
1	F	F	F	F	F
2	F	F	F	F	F
3	F	F	F	F	T
4	F	F	F	F	T
5	F	F	F	F	T
...
20	F	F	T	F	T
21	F	F	F	F	T
...
100	F	F	F	T	T



[그림 5] 유사 환경인지 실험결과 2

[표 2]와 [그림 5]에서 보듯이 대상 에이전트에 대한 학습정보나 다른 에이전트의 학습정보가 없으면 비교대

상이 존재하지 않기 때문에 유사한 환경을 찾지 못한다. 하지만 ED+Q-Learning을 통해 주변 에이전트들의 학습정보 보다는 대상 에이전트의 학습정보가 많으면 많을수록 유사한 환경을 빨리 찾는다라는 것을 알 수 있다.

5. 결론

기존에 다중 에이전트 환경에서 여러 에이전트들이 단일 환경에 존재하는 상태에서 학습속도 향상을 위해서는 여러 방법들이 제안되었다.[4][6][7]. 그러나 본 논문에서는 기존의 단일 에이전트가 학습할 때 발생하는 문제점과 다중 에이전트 환경에서 각 에이전트들이 서로 다른 환경에 존재할 때 나타나는 문제점을 해결하기 위해 편집거리를 사용한 ED+Q-Learning 시스템을 제안하였다. 제안한 ED+Q-Learning 시스템은 다중 환경에 위치하는 에이전트들 간에 학습정보를 사용하기 위해 편집거리로 에이전트간의 유사도를 측정하고 유사한 에이전트의 정보를 Q-Learning에 이용할 수 있도록 하였다. 그럼으로써 Q-Learning의 단점인 거대한 환경 및 초기 환경에서 학습속도가 저하되는 문제점을 해결하였다.

국문 : 본 논문은 정통부 및 정보통신연구진흥원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.

참고 문헌

- [1] Mitchell, T. M., "Machine Learning", McGraw-Hill, 1997.
- [2] 최종민, "에이전트 개요와 연구방향", *정보과학회지*, 제 15권, 제 3호, 7-16쪽, 1997.
- [3] Peeters, M., "A Study of Reinforcement Learning Techniques for Cooperative Multi-Agent Systems", Master's thesis, 2003.
- [4] Ito, A., "Application of Episodic Q-learning to a Multi-agent Cooperative Task", *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence*, pages 188-197, Tokyo, Japan, 2002.
- [5] Schoknecht, R., "Hierarchical Reinforcement Learning with Multi-step actions", In *Proceedings of the 18th International Conference on Machine Learning*, 2001.
- [6] Rajbala Makar, Sridhar Mahadevan, and Mohammad Ghavamzadeh, "Hierarchical Multi-Agent Reinforcement Learning", *Fifth International Conference on Autonomous Agents*, Montreal, 2001.
- [7] M. Nili Ahmatabadi, Masoud Asadpur, Seyyed H. Khodaabakhsh, and Eiji Nakano, "Expertness Measuring in Cooperative Learning", *Proc. IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*, Japan, 2000.

웹 문서 수집을 위한 효율적인 문서 분류

이정훈^o 전서현 *김선희
동국대학교 컴퓨터공학과
*전남대학교 전산학과

{leeye123^o, shcheon55, wkdal749}@dgu.ac.kr

Efficient Document Classification for Web Document Collection

Jung Hun Lee^o Suh Hyun Cheon *Sun Hee Kim
Department of Internet Computing, Dongguk University
*Dept. of Computer Science., Chonnam National University

요 약

최근 다양한 형식의 웹 문서에서 사용자가 원하는 정보만을 검색 하기위해 웹 문서를 주제별로 분류하여 수집하고, 관리하는 것은 필수적인 요소이다. 즉, 정확하고 빠른 정보 검색을 위한 웹 문서 수집은 문서 형식에 따라 분류되어 수집 되어야 한다. 따라서 웹 환경에서 문서를 구성하는 형식을 텍스트나 이미지 데이터로 구분하고 그 형식에 맞는 분류기법을 사용한다면 정확한 정보 검색이 이루어 질수 있다. 본 논문에서는 텍스트와 URL을 이용한 주제 중심의 하이브리드 웹 문서 분류 방법을 제안한다. 텍스트와 URL을 이용한 분류 방법은 텍스트 형식은 주제 중심의 문서 분류방식을 사용하며, 텍스트 정보의 효용성이 낮은 경우 URL의 주제 분포도를 이용하여 분류하며 수집한다. 이를 통해 여러 가지 형식의 웹 문서가 분류 가능하며, 주제에 따른 문서 분류의 정확도가 높아진다.

가진다.

1. 서 론

웹 문서는 초기에 텍스트 형식의 웹 문서에서 효과적인 정보전달을 위하여 이미지나 멀티미디어와 같은 다양한 형식의 웹 문서 형식들로 변화 하면서 다양한 형식의 웹 문서를 접할 수 있게 되었다. 웹 문서의 형식이 변화 하면서 웹 문서에서 자신이 원하는 정보를 찾아내는 것은 점점 어려워지고 있다. 일반 사용자들은 자신이 원하는 정보를 얻기 위해 야후(Yahoo)나 네이버(Naver)등의 검색엔진을 사용하지만 원하는 정보를 찾기까지는 많은 시간과 노력이 소요된다. 또 무분별한 웹 문서의 수집으로 검색엔진은 적합한 결과물을 제공해 주지 못하고, 방대한 데이터로 인해 검색 엔진의 부하가 생기게 되었다.

이러한 문제를 해결하기 위해 검색 결과를 종합하여 보여주는 메타 서치 엔진[9,10], 텍스트 데이터를 이용한 텍스트 기반의 주제 중심 검색에 대한 연구가 수행 되었다[5]. 그러나 기존의 검색 방식은 모두 텍스트 형식의 검색 방식으로 다양한 형식을 가지는 웹 문서를 분류하거나 필터링 하기는 어렵다는 단점이 있다. 또 다른 연구로 텍스트 형식의 검색 방식이 가지는 단점을 보완하기 위해 하이퍼텍스트간의 상호 관련성을 이용하여 분류하는 방식이 연구 되었다[8]. 그러나 하이퍼링크의 상호 관련성 분류 방식은 웹 문서의 내용을 고려하지 않아 질의어와 관련 없는 내용의 문서를 잘못 분류하고, 관련된 문서에 하이퍼링크가 없을 경우 분류를 할 수 없게 되는 문제점을

일반적으로 웹 문서들은 텍스트와 이미지 또는 멀티미디어와 같은 다양한 형식으로 구성된다. 따라서 여러 가지 형식을 고려한 웹 문서 분류기법을 적용한다면 검색의 정확성과 효율성을 높일 수 있다.

본 논문에서는 복잡한 형식의 문서를 고려하여 사용자가 원하는 정보를 정확하게 검색할 수 있는 하이브리드 웹 문서 분류 기법을 제안한다. 일반적으로 웹 문서의 텍스트 정보를 이용한 텍스트 기반의 주제 중심 분류 기법을 통하여 분류를 하고 분류 가능한 텍스트 데이터가 부족할 경우 하이퍼링크의 주제 분포도를 이용하여 분류한다.

본 논문의 2장에서는 웹 문서 형식에 따른 분류기법을 살펴보고, 3장에서는 하이브리드 분류 기법을 제시한다. 4장에서는 제안한 기법의 정확성을 측정하기 위한 실험 방법과 실험결과에 대하여 분석하고, 5장은 결론과 향후 연구 과제에 대하여 기술한다.

2. 관련 연구

주제를 중심으로 웹 문서를 분류하는 분류기법에 대한 연구는 크게 두 가지 방향을 나누어 볼 수 있다. 첫 번째는 텍스트 정보를 이용한 텍스트 기반의 주제중심 분류 기법이며, 두 번째는 하이퍼링크를 이용한 주제 중심의 분류 기법이다.

2.1. 텍스트 정보를 이용한 웹 분류

웹 문서 분류 기법은 수집된 웹 문서를 재배열하여 주제와 부합되는 웹 문서만을 수집하는 방식에서 시작하였다. SharkSearch[3]는 문서의 연관성을 판단하기 위해 코사인 유사도 측정을 위한 TF(Term Frequency) 또는 IDF(Inverse document Frequency)와 같은 전형적인 정보 검색(Information Retrieval) 알고리즘을 사용하였다. 그러나 SharkSearch 알고리즘은 분류기를 이용하지 않는 모델로 단순히 유사도를 측정하기 위해 정보검색 기반 기술을 이용하였다. Charabati[6]는 주제를 중심으로 웹 문서를 수집하는 방식에서 분류기를 처음으로 사용하여 불필요한 웹 페이지 방문을 차단하여 문서를 수집하는 시간을 줄였다. 하지만 Charabati는 관련 있는 문서로 평가된 웹 문서들의 하이퍼링크들은 고려하지 않고 있다. Cora[1]는 최적화 기법인 강화 학습 알고리즘을 이용하여 수집경로를 결정하였다. 그러나 이러한 연구들도 수집 경로를 구성할 때 관련 있는 문서로 평가된 웹 문서를 통해 작업이 이루어짐으로 관련 없는 문서로 평가되었지만 문서에 내재된 연관성 있는 하이퍼링크들은 고려하지 않고 있다.

2.2. 하이퍼링크를 이용한 웹 분류

하이퍼링크는 다른 문서를 연결하기 위한 것으로 하이퍼링크의 상호 참조 관련성 정보를 이용해 하이퍼링크가 관련 있는 문서를 가리키고 있는지 판단할 수 있다. FishSearch[4]는 특정 URL을 시발점으로 주어진 질의에 적합한 내용을 가진 문서를 고려하여 방문할 URL의 목록에서 우선순위에 따라 수집하는 형식으로 접근하였다.

URL의 ordering을 통한 Crawling[2]에서는 사용한 문서가 다른 문서를 연결하고 있는 수(in-links), 문서 링크(page-link)와 같은 경험적인 방법이 제시되었다. 이 방식으로 문서 수집 시 우선순위에 따른 수집 순서를 재정의하여 연관성 있는 문서를 먼저 수집 될 수 있도록 하였으나 분류기를 이용하지 않은 초기 모델이며, 이 방식은 단순히 유사도를 측정하기 위하여 사용되었다. 하이퍼링크는 하이퍼링크의 상호 참조 관련성을 이용하여 관련 있는 문서를 분류 할 수 있다. 그러나 방대한 양의 텍스트 형식일 경우 웹 문서의 URL을 모두 고려할 수 없다. 따라서 너무 많은 텍스트로 문서가 구성된 경우 오분류율(misclassification rate)이 높아진다.

따라서 본 논문에서는 일반적으로 가장 많이 사용되고 성능이 입증된 Naïve Bayes[7] 확률 모델을 이용하여 관련성 여부를 판단하여 텍스트나 이미지와 같은 다양한 형식으로 구성된 문서의 오분류율을 줄인다.

3. 하이브리드 방식의 웹 문서 분류기

본 논문에서 제안하는 하이브리드 분류기는 텍스트 정보

와 하이퍼링크를 이용한 주제 중심의 분류 기법으로, 그림 1은 하이브리드 방식 분류기의 전체적인 알고리즘이다.

```

Uri = Get_Input_start_url();
HostName = Get_HostName_Parsing(Uri);
Document_Resemblance_rate = Get_Document_resemblance_rate();
Document = Get_Web_page_Document();
Document_Classification = TRUE;

if(MINIMUM_DOCUMENT_RESEMBLANCE_RATE <= Document_Resemblance_rate)
{
    Naive_Bayesian_Classification(Document);
}
else
{
    HostName_resemblance_rate = Get_HostName_resemblance_rate(HostName);
    if(MINIMUM_HOSTNAME_RESEMBLANCE_RATE <= HostName_resemblance_rate)
    {
        Uri_Classification(Uri);
    }
    else
    {
        Hyperlinks_Number = Get_Document_Hyperlinks_Url_Number();
        while(Hyperlinks_Number > 0)
        {
            Hyperlinks_Number--;
            Child_Url = Get_Document_Hyperlinks_Url;
            Child_Url_HostName = Child_Url_HostName + Get_HostName_Parsing(Child_Url);
        }
        HostName_resemblance_rate = Get_HostName_resemblance_rate(Child_Url_HostName);

        if(MINIMUM_HOSTNAME_RESEMBLANCE_RATE <= HostName_resemblance_rate)
        {
            Uri_Classification(Uri);
        }
        else
        {
            Document_Classification = FAILURE;
        }
    }
}

return Document_Classification;
    
```

그림 1. 하이브리드 방식 분류기의 Pseudo code

그림 1과 같이 하이브리드 방식은 웹 문서를 분류하기 위해 텍스트 정보를 이용한 분류와 하이퍼링크 이용한 두 가지 분류 단계로 구성되어 있다.

첫 번째는 웹 문서의 텍스트 데이터를 추출하여 효용성을 측정한다. 텍스트 데이터의 효용성은 웹 문서의 텍스트 데이터의 빈도수를 이용하여 전체 문서에서 텍스트 데이터가 차지하는 비율을 식 1과 같이 계산한다.

$$Utility(x) = \frac{1}{Length(x)} \sum_{k=1}^{|x|} TF(w_k, x) \quad (1)$$

Length(x)는 문서 x가 가지는 전체 텍스트의 수를 나타내며, TF(w_k, x)는 x에 포함된 각 단어의 빈도수를 나타낸다. 문서의 효용성을 측정하여 65%이상일 경우는 텍스트 정보를 이용하여 텍스트 기반의 주제 중심으로 웹 문서를 분류하고, 65% 이하일 경우 하이퍼링크로 연결된 웹 문서의 텍스트 정보를 추출한다. 추출된 정보는 다시 효용성을 측정하여 65% 이하일 경우, 두 번째 단계로 하이퍼링크의 주제 분포도를 이용한 웹 문서를 분류한다. 그림 2는 웹 문서 분류를 위한 효용성 및 문서유사도 측정 알고