

## RTI 통신을 이용한 개인환경기반 자동문서 분산처리 기술

인주호\*, 김명규, 채수환  
한국항공대학교 컴퓨터공학과

## A Distributed Processing Model for Automatic Classification of Text Documents based Personalized Information Using RTI

JooHo In\*, MyungKyu Kim, SooHoan Chae  
Dept of Computer Engineering, Korea Aerospace University  
{allpoyou, kimmk, [chae](mailto:chae@hau.ac.kr)}@hau.ac.kr

## ABSTRACT

인터넷이 폭 넓게 보급되어 온라인 상에서 얻을 수 있는 텍스트 정보의 양이 급증함에 따라 산재해 있는 문서들에 대한 효과적인 정보 관리 및 검색이 요구되고 있다. 자동 문서분류란 문서의 내용에 기반하여 미리 정의되어 있는 범주에 문서를 자동으로 할당하는 작업으로써 효율적인 정보 관리 및 검색을 가능하게 한다. 하지만 자동문서 분류를 하기 위해서는 방대한 양의 데이터를 수집 보관하기 위한 분산 환경이 반드시 필요하다. 본 논문에서는 자동 문서 분류를 위한 분산기반 환경의 조성에 있어서 RTI(Run Time Infrastructure)를 통한 분산 시스템 환경으로 구성하였다.

## 1. Introduction

최근에는 신문이나 잡지와 같은 미디어부터 인터넷을 이용한 전자매체까지 다양한 경로를 통해 정보를 습득할 수 있게 되었다.

특히, 인터넷의 확산과 더불어 전자매체를 이용함으로써 방대한 양의 정보를 통합하여 사용자에게 제공함으로써 보다 편리하게 정보를 얻고 활용할 수 있게 되었다. 이와 같이 온라인상에서 얻을 수 있는 정보가 기하급수적으로 급증함에 따라 효율적인 정보관리 및 검색이 요구되고 있다.

이에 따라 정보습득 작업을 자동 분류 시스템을 사용하면 사람이 직접 손으로 분류할 경우보다 비용을 크게 줄일 수 있으며, 자동 문서 범주화는 대량의 문서를 효율적으로 관리하고 검색할 수 있게 하는 동시에 방대한 양의 수작업을 감소시키는 데 그 목적이 있다.

또한 한국어 정보처리에 관한 수많은 개발자와 연구자들이 관련분야의 자료들에 대한 효율적인 수집 및 분류를 통한 정보습득, 일반 인터넷사용자들의 정보수요구 등이 날로 급증하고 있어 이 또한 정보관리 및 검색이 요구되고 있다.[1]

하지만 단일 서버에서 대량의 문서를 자동으로 범주화 및 분류를 시행하는 것은 시스템의 성능이라는 제약 조건을 가지게 된다. 이에 따라 시스템의 성능을 분산시키고 효율적인 데이터 관리를 위한 분산처리기술의 개발은 필연적으로 요구되는 바이다.

따라서 자동문서 분류 수행단계에서, 한글 웹 문서들의 각 분야에 대해서 사전구축을 시행하고 이를 바탕으로 개별 서버에 주제를 나누어 한 개의 서버가 한 개의 주제를 관리할 수 있게 하여 문서 분류/관리의 성능적 향상을 도모한다.

## 2. Related Works

## 2.1 자동 문서 분류의 개관

자동 문서분류는 일반적으로 주제어 추출 과정과 분류과정으로 나눌 수가 있으며, 주제어추출과정은 전처리 과정과 주제어추출과정을 거쳐 문서에 출현하는 단어들을 바탕으로 문서를 재형성한다. 전처리 과정은 문서로부터 태그와 불용어를 제거하고 형태소 분석을 통해 특정 용어들을 추출하게 된다. 주제어추출과정은 형태소분석을 통해 나온 특징들 중 가치가 있는 명사들로만 추출하는 과정과 함께

가중치를 결정하기 쉽게 하기 위해 특정방법으로 정규화하는 과정을 말한다. 단어에 대한 가중치를 계산하는 방법은 이진 주제어, 단어 빈도, 역문서 빈도, 단어 빈도와 역문서 빈도의 곱 등이 있다. [2,3,4] 본 논문에서는 단어빈도와 편차 그리고, 정규화 방법을 이용하였다.

그림1은 전체적인 자동 문서분류를 모델화 한 것이다.

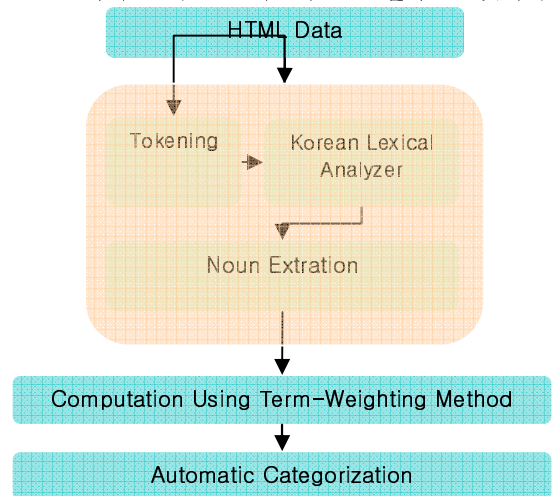


그림 1. 자동문서분류 모델

## 2.1.1 자동 문서 분류 방법

문서 자동 분류 방법에는 통계적 방법과 의미 분석방법으로 구분할 수 있다. 후자의 경우 자연어 자체의 모호성 때문에 그 사용이 어렵고 한정되어 있는 반면, 통계적 방법은 간단히 구현할 수 있고 학습이 가능하여, 충분한 학습 데이터가 주어졌을 경우 의미 분석방법에 버금가는 결과를 낼 수 있다.

통계적 문서 분류 방법에서는 문서를 대표하는 용어와 이것의 가중치를 결정하는 방법이 필요하다, 용어의 가중치의 계산에 용어의 문서 내 빈도를 고려하는 경우가 있는데, 이것을 용어빈도 가중치라고 한다.

용어빈도 가중치 계산방법은 문헌 내 출현 여부만을 반영하는 이진 값이나 출현빈도 자체를 가중치로 사용할 수도

있으며, 이와외 다양한 공식이 제안되었다. 이렇게 용어에 가중치를 부여함은 한 문서가 취급하고 있는 개념들의 주제적 요소로서의 중요도에 따라, 색인어로서 상대적 가치를 표현하기 위함이다. [1, 5, 6]

본 논문에서는 각 분야에 대한 사전을 구축하고, 웹 문서의 가중치를 계산하는 방법으로 이 가중치 계산방법을 적용과 함께 정규 사전을 사용했을 때와의 문서분류의 차이를 알아보기 위해 주제범주간 기사들간의 용어정보분석을 수행하였다.

## 2.2 RTI (Run Time Infrastructure)

RTI는 운영체제와 federate들 사이에 있는 미들웨어 소프트웨어로 각 federate들의 데이터 교환 및 시뮬레이션 시간 진행에 필요한 여러 가지 기능을 제공한다. RTI는 다음과 같은 세 가지로 구성되어 있다[11].

### (1) RTIExec (RTI Executive Process)

Federation 실행의 생성과 소멸을 관리하는 전역 프로세스로, FedExec가 서로 다른 이름을 갖도록 하며 수동 조작을 위한 인터페이스를 제공하는 역할을 수행한다.

### (2) FedExec (Federation Executive process)

실행중인 federation에서 생성되는 하나의 프로세스로, 생성된 federation을 관리하면 federation에 참가하는 federate들에게 핸들을 할당하여 federation 실행에 참가하고 탈퇴하는 것을 관리한다.

### (3) libRTI (RTI Library)

C++ 라이브러리인 libRTI는 응용 프로그램에 포함되는 클래스 라이브러리로, 각각의 응용 프로그램들을 libRTI를 통해서 RTI의 서비스를 호출할 수 있다[12].

Federation의 실행은 그림 2처럼 3단계로 수행되어진다. 각 단계는 아래와 같다.

- (1) 사용자는 RTI를 시작하여 RTIExec 프로세스를 실행시켜 새로운 federate가 참가 가능 하도록 준비한다.
- (2) 사용자는 federate를 실행하고 이 federate는 새로운 federation을 만들며 FedExec 프로세스를 실행시킨다.
- (3) 새롭게 추가되는 federate들은 FedExec를 통하여 기존의 federation에 참여하게 된다.

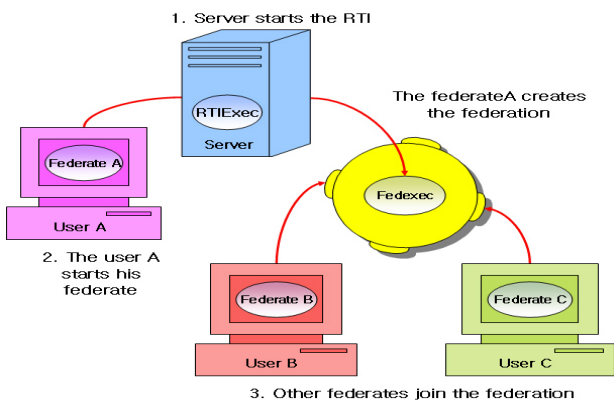


그림 2. RTI federation 의 진행

RTI, 상호운용성을 위한 본질적인 방법을 제공한다. 이것은 federate들과 federation간의 정보교환을 가능하게 하고 실시간 데이터 전송에 관한 수단과 개념을 제공한다[14] [15] [16].

이를 통해 RTIExec는 각 federation들에게 데이터를 전송함에 있어서 특정 위치를 정하는 것이 아닌 RTI에 데이터를 전송하고 모든 federation들은 RTI로부터 데이터를 얻어 낼 수 있다.

## 3. 실험환경 모델

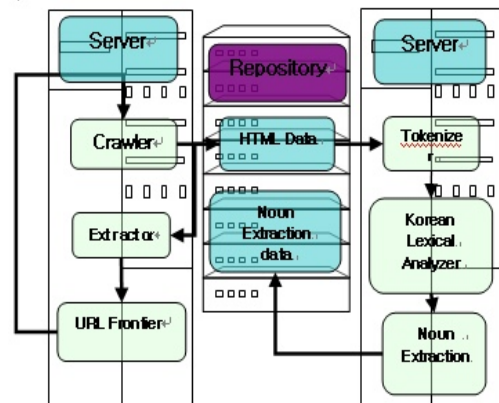


그림 3. Subject Extraction System

[그림 3]은 자동 문서분류 과정 중 첫 번째 단계로 주제어 추출모델을 나타낸다. 한글 웹문서를 대상으로 각 문서를 대표할 수 있는 주제어를 추출 해낸다.

### 3.1.1 Crawler

그림4는 구현된 Crawler의 System Architecture를 나타낸다.

크로울러는 seed URL 주소로부터 시작하여 자원이 사용 가능한 한 계속적으로 페이지들을 방문하고 새로운 주소를 추출 후 넘겨받아 넘겨받은 주소를 다시 방문하는 과정을 반복한다[13]

Extractor는 다운 받은 html 파일로부터 Child URL들을 추출하게 되고 Frontier에 중복된 URL이 있는지 확인하는 DUE를 통해, 없으면 URL Frontier에 Child URL들을 저장하게 된다. [5,6,7]

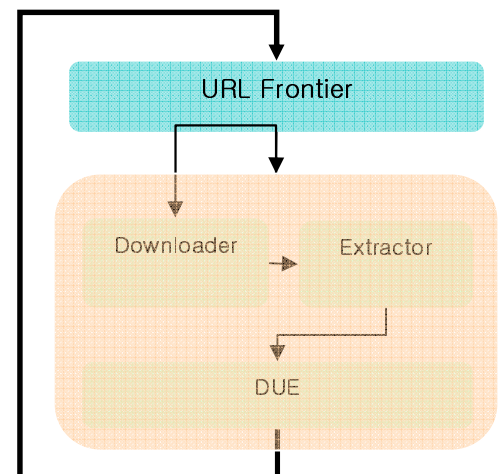


그림 4. Crawler System Architecture

### 3.1.2 Repository, Tokenizer

Repository는 Crawler를 통해서 다운 받은 한글 웹 문서들을 Html형식으로 저장하는 공간이다. Tokenizer는 Repository를 통해 저장된 파일에서 태그 제거 및 contents추출을 시행하여 파일의 내용을 token화 하여 txt파일로 재형성하는 역할을 한다.

### 3.1.3 Korean Lexical Analyzer

Token화된 웹 문서에 대해서 그 문서를 대표 할 수 있는 한글 주제어들을 추출하기 위해 형태소분석기(HAM)를 사용하였다.

한국어는 영어와 달리 한 어절이 하나 이상의 형태소로 이루어져 있기 때문에 주제어를 추출 하기 위해서는 어절을 이루고 있는 형태소들에 대한 인식 및 분리가 이루어져야 한다.[4]

### 3.1.4 Noun Extraction , Noun Extraction data

Noun Extraction은 형태소 분석한 결과에서 명사데이터 만을 추출해내는 역할을 한다. 이후 명사만으로 이루어진 파일을 Noun Extraction data에 저장하게 된다.

## 3.2 주제어 축소

사전을 구축하고 효율적인 분류와 불필요한 연산을 줄이기 위해 각 문서를 대표하는 주제어 들에 대한 축소과정이 필요하다.

각 문서에 포함되어 있는 주제어들을 추출하고 각 단어에 대해서 형태소분석을 통한 주제어로서 가치가 있는 명사만을 추출한다.

## 3.3 사전 구축

본 논문에서는 각 한글 웹문서에 대한 가중치를 계산하는 방법으로 각 분야별 사전을 구축하는데, 주제어를 추출하고, 축소하는 과정은 모든 문서 에 대해서 수행된다.

사전을 작성하는데 있어서는 대표하는 용어 들이 사전에 너무 많거나 적은 것은 주제어로서 적당하지 않고, 적정수준의 용어들이 주제어로 적당하다는 직관적인 논리를 적용했다.

이를 통해 중복주제어제거기를 통해 각 분야별 중복된 주제어를 나타내는 임계치만큼 중복 제거를 하여, 각 사전의 독립성을 높일 수 있었다.

## 3.4 Categorizator

Categorizator는 가공된 각 한글 웹문서와 구축된 각 분야별 사전을 통해서 주제어들의 빈도 ( frequency )를 계산하여, 가장 높은 weight를 가진 분야로 범주화하는 기능을 가진다.

그림 5는 각 에이전트에서 실시되는 범주화 기능의 모델을 제 시한 것이다.

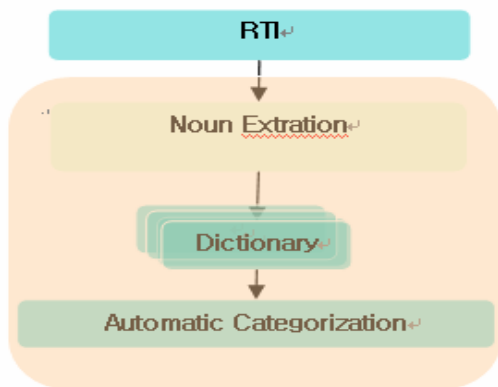


그림 5. Categorizator Architecture

## 3.5 RTI 기반 분산환경 구축

앞서 제기한 전처리 시스템부터 주제어 축소까지의 과정을 거친 문서들은 단일 시스템이 아닌 분산 시스템에 그 데이터를 나누어서 관리하게 된다. 이 과정에서 각 분산 시스템에서 주제별 사전을 가지고 있는 agent를 통해서 들어오는 데이터의 범주를 계산하여 해당 주제의 문서정보만을 모아서 관리하게 된다. 이때 각 federate들은 다운 받은 웹 페이지를 관리 하는 것이 아닌 웹 페이지가 저장장치인 RAID 어느 곳에 위치해있는지에 관한 위치정보와 페이지의 범주를 계산하기 위한 추출된 주제어 정보만을 받게 된다.

데이터 범주의 계산은 3.4절에서 논한 Categorizator의 과정을 통해서 행하게 된다. 각 문서의 분류 시 통신을 위해 RTI를 활용 한다. RTI는 앞서도 말했듯이 federation간의 실시간 데이터 전송에 관한 수단 과 개념을 제공한다 [14] [15] [16]. 그림 6은 이상의 시스템의 전체적인 구성을 보여준다.

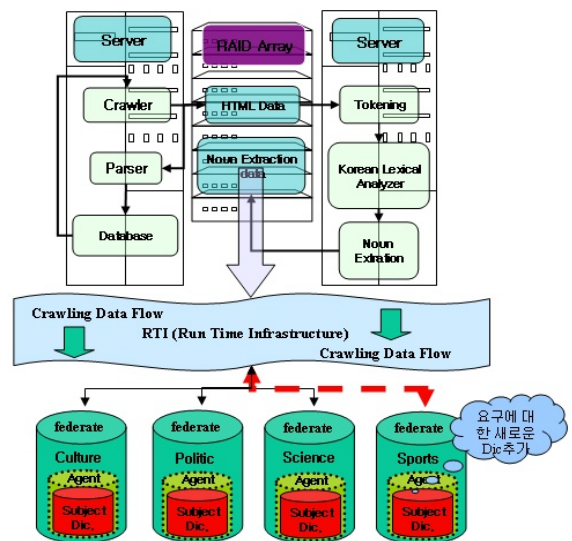


그림 6. Distributed Document Management System with RTI

## 4. 구현 및 실험 분석

### 4.1 실험 방법 및 데이터 구성

실험은 IT정규사전을 통한 문서 분류의 성능을 측정 후 개별적인 주제의 사전을 에이전트(agent)화 하여 다양한 주제를 각 agent에서 자동으로 걸러내는 것을 실험 하였다. 문서 분류의 실험 문서집단으로서는 한겨레 신문의 문화, 정치 분야 300건, 전자신문의 정보통신 분야 349건의 기사를 이용하였다. 이 중 각 카테고리당 100 건의 기사를 Train Set으로 하여 사전구축을 하였고, 나머지 기사들을 Test Set으로 두어 실험하였다.

실험에 사용된 문서를 대표하는 용어들의 대표성, 즉 범주화에 사용될 각 분야별 weight값들을 결정하기 위해 각 분야별 사전을 구축한 Train Set 기사들의 분석이 필요했다.

표1은 각 분야의 사전구축에 필요했던 기사 들의 주제어종류와 수 및 IT정규 사전의 주제어 수를 나타낸다. 나타난 것이다. 정규 사전은 네이버의 IT사전 인덱스를 인용하였다.

표 1 Information on paper and Dictionary

	Dictionary(사전)	IT(기사)	Culture&Politic
Total	22740	48708	41358
주제어종류	22740	4467	4080

## 4.2 RTI통한 분산처리

다음의 그림은 RTI를 통한 federation들의 연결 화면이다.

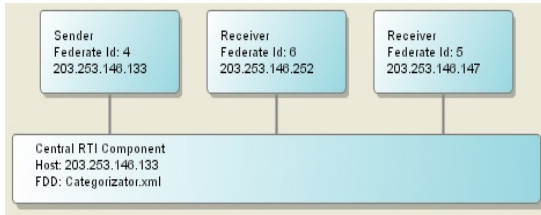


그림 7. Federation Connection

각 federation들은 Crawler서버인 Sender로부터 웹데이터의 위치와 주제어 정보를 받는다. 이후 각 federation에 설치된 Categorizator를 통해 해당 주제의 분류를 시행 federation별 해당 주제에 관한 데이터만을 저장 관리 한다. federation에서 다루지 않는 주제에 대해서는 federation에서는 Garbage처리를 통해 관리 대상에서 제외 시킨다.

IT주제의 경우 사전 정규화의 실험 군으로서 정규 사전을 가지고 있는 agent와 기사를 통해 구축한 사전을 가지고 있는 두 가지 agent를 두었다. 이 두 agent에 RTI를 통해 데이터 전송 후 두 agent에서 IT관련 기사를 분리해 내는 실험을 먼저 진행하였다.. 이를 통해 각 주제 사전을 통한 문서분류의 시행에서 각 federation들이 해당 주제의 데이터 들만을 추출해 낼 수 있다. 또한 기사의 주제어를 통해 사전을 만든 통계적 문서분류 방식의 정확성이 정규사전을 이용한 것에 근접한 결과를 보임을 알 수 있다. 그림 8는 IT사전을 적용했을 때의 문서 분류 결과이고 8.a는 IT기사를 통해 구축한 사전을 적용해 분류한 결과 화면이다.

표 2는 IT정규 사전과 기사를 통해 구축한 사전으로 IT분야의 문서 249건과 기타주제 300건을 RTI를 통해 전송 이들의 분류 결과를 나타낸 것이다. 무시된 항목은 사전 적용 결과 IT항목에 적합하지 않다고 판단하여 자동분류 agent에서 Garabage항목으로 분류된 기사의 건수이다. Miss는 IT관련 기사가 Garabage로 잘못 처리된 건수, 실패(fail)은 Garabage로 처리되어야 할 문서가 IT항목으로 받아들여진 경우의 수를 나타낸다.

```

=====
politic :0.0 , eco :0.07446808
sports :0.010638298 , culture :0.0 , IT :0.7234042
This page's biggest weight is ITweightValue => 0.7234042

=====
politic :0.0 , eco :0.028571429
sports :0.028571429 , culture :0.0 , IT :0.71428573
This page's biggest weight is ITweightValue => 0.71428573

=====
politic :0.0 , eco :0.0
sports :0.0 , culture :0.0 , IT :0.59574467
This page's biggest weight is ITweightValue => 0.59574467
249
0
===== IT Categorization Result Value =====
IT :0.9799197
Garbage :0.0
=====

```

그림 8. 범주화 계산 과정

```

=====
politic :0.0 , eco :0.07446808
sports :0.010638298 , culture :0.0 , IT :0.05319149
This page's biggest weight is ecoweightValue => 0.07446808

=====
politic :0.0 , eco :0.028571429
sports :0.028571429 , culture :0.0 , IT :0.028571429
This page's biggest weight is ecoweightValue => 0.028571429

=====
politic :0.0 , eco :0.0
sports :0.0 , culture :0.0 , IT :0.14893617
This page's biggest weight is ITweightValue => 0.14893617
249
1
===== IT Categorization Result Value =====
IT :0.9196787
Garbage :0.056224898
=====

```

그림 8.a. 범주화 계산 결과

표 2 정규사전과 기사를 통해 구축한 사전의 분류 결과

	Dictionary(사전)	IT(기사)
IT로 분류된 기사	320	246
Garabage로 분류된 기사	329	303
miss	4	5
fail	33	2

사전을 통한 문서 분류 시 기사 구축을 통한 사전 보다 좀더 많은 기사를 Garabage처리 하였다. 이는 문서와 사전간의 적중률(hit rate)계산시 Dictionary의 어휘수가 기사를 통해 구축한 사전보다 더 많은 어휘를 가지고 있음으로 해서 짧은 기사의 전체 사전 어휘대비 hit rate가 너무 낮음으로 해서 생기는 문제이다. 이에 대해서는 문서분류 시에 기준 가중치 부여에 있어서 수치 조정이 필요하다. 위의 결과를 통해서 IT agent 에서의 분류 과정을 검토해 보면 IT주제를 가진 문서를 거의 정확하게 분류해낼 수 있다. 이 과정을 확장하여 정치와 사회에 관한 사전을 구축 RTI를 통해 들어오는 문서의 분류는 위의 표와 거의 비슷한 수치들을 보이고 있다. 아래의 그림은 위의 과정을 확장하여 RTI에 연결된 federation중 하나에 정치주제를 부여하고 이를 분류한 것에 대한 결과이다.

```

=====
politic :0.07883818 , sports :0.0013831259 , culture :0.0027662518
This page's biggest weight is politicweightValue -> 0.07883818

=====
politic :0.084078714 , sports :0.0 , culture :0.0017889087
This page's biggest weight is politicweightValue -> 0.084078714

=====
politic :0.22693096 , sports :0.0 , culture :0.0
This page's biggest weight is politicweightValue -> 0.22693096

=====
politic :0.2112403 , sports :0.0 , culture :0.004844961
This page's biggest weight is politicweightValue -> 0.2112403
83
0
===== Politic Categorization Result Value =====
politic :0.48192722
Garbage :0.5180723
=====

```

그림 9. 정치분야 기사 분류 결과 화면

그림 10은 분류해야 할 데이터의 주제가 늘어나는 경우 RTI를 통한 agent확장화면이다. 임의의 장비에 분류하고자 하는 주제의 사전을 구축하고 RTI federation에 연결함으로써 추가적인 주제의 문서 분류를 시행 함을 확인 할 수

있다.

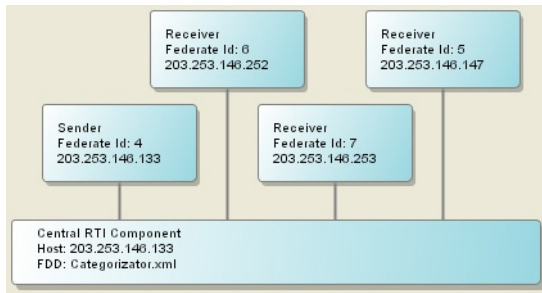


그림 10. Extend Federation Connection

## 5. 결론 및 Future Work

본 논문에서는 분야별 사전을 구축하는데 기반이 되는 웹 기사에 대해 분석을 하고 그 결과를 토대로 사전을 만들어 각 federation에 agent 설치 이를 바탕으로 주제별 분산 분류를 시행 했다. 각 federation들은 RTI를 통해 웹 페이지의 저장장치에서의 위치정보, 그리고 해당 문서의 주제를 축소한 결과 받을 바에서 이를 통해 갖추어진 사전과 비교해 문서분류를 분산 처리를 할 수 있었다. 각 federation들에게는 파일 전체가 아닌 비교에 필요한 요약 정보만을 줌으로써 각 federation이 파일을 가지고 직접 비교 할 때의 경우보다 많은 로드를 줄일 수 있다. 또한 모든 실제 데이터는 RAID array한곳에서 관리하기에 각 federation들의 디스크공간의 중복 낭비 방지의 효과도 기대할 수 있다.

전제적인 시스템에서 문서분류의 정확성은 각 federation들의 사전을 얼마나 정확성을 높여서 구축하느냐에 있다.

향후 구축에 필요한 주제 범주간의 기사 별 용어정보에 대한 분석을 좀더 정확성 있게 시행하고, 정규 용어 사전에 준하는 분야별 사전 구축을 통해 보다 효율적인 자동분류를 수행할 수 있는 시스템을 만들기 위한 방법이 필요하다.

## Acknowledgements

"본 논문은 산업자원부 한국산업기술평가원 지정 한국 항공 대학교 부설 인터넷정보 검색 연구센터의 지원에 의함"

## References

- [1] Go young jung, Seo jeong yeon, "Theory and Techniques about automatic categorization for document management", 2002
- [2] Bao, Y. and Ishii, N., "Combining Multiple K-Nearest Neighbor Classifiers for Text Classification by Reducts", In Proceeding of the fifth International Conference on Discovery Science, 2002
- [3] Sasaki, M. and Kita, K., "Rule-Based Text Categorization Using Hierarchical Categories", In Proceeding of the IEEE International Conference on System, Man and Cybernetics, 1998
- [4] Sebastiani, F., "Machine learning in automated text categorization", ACM Computing Survwys, 2002
- [5] Jaeyun, lee., Boyeong, Choi., yeongmi, Jeong., " Research about term weight's techniques in literature automatic categorization", Korea Society for Information Management, 2000
- [6] Gwangje, Jo., Juntae, Kim., "Research document's automatic categorization in hierarchic category system by Inverted Term Frequency", Korea Information Science Society, 1997

- [7] Marc Najork and Allan Heydon, "High-Performance Web Crawling", 2001
- [8] Gautam Pant, Padmini Srinivasan and Filippo Mnecher, "Crawling the Web", 2003
- [9] Soumen Chakrabarti, Mining the Web. Indian Institute of Technology, 2003,
- [10] Seungsik, Kang., Korean Morpheme Analysis and Information Retrieval, Hongreung Science Publishing Company, 2002
- [11] 서혜숙, 한상범, 신중희, 황종선, "XML을 적용한 HLA 기반 시뮬레이션" 한국정보과학회 2003 춘계학술발표논문집, pp.115-117
- [12] 차영필, 정무영, "분산 생산 시스템을 위한 agent기반의 협업 시뮬레이션 체계" 2003 춘계공동학술대회.
- [13] A. Arasu, J.Cho, H.Garcia-Molina, A.Paepcke, S.Raghavan, "Searching the Web", ACM Transactions on Internet Technology, Vol.1, Num. 1, August 2001, pp.2-43.
- [14] IEEE 1516, "Standard for Modeling and Simulation High Level Architecture - Framework and Rules" IEEE, 2000.
- [15] IEEE 1516.2, "Standard for Modeling and Simulation High Level Architecture - Federate Interface Specification" IEEE, 2000.
- [16] IEEE 1516.1, "Standard for Modeling and Simulation High Level Architecture - Object Model Template Specification" IEEE, 2000.