

퍼지 필터링 기반의 메시지 사서함 서비스를 위한 genealogy 그룹화

Genealogy grouping for services of message post-office box based on fuzzy-filtering

이종득* · 안정용**

Chong-Deuk Lee* · Jeong-Yong Ahn**

* 국립익산대학 정보통신공학과

** 전북대학교 통계정보학부

요 약

사서함 구조에서 메시지 서비스를 위한 중요한 구조적 메커니즘은 메시지 객체들을 콘텐츠에 따라 그룹 클래스 계층 구조를 구성하는 것이다. 본 논문에서는 응용 도메인에서 많은 구조적인 메시지 객체들을 클러스터링하기 위한 α -cut 기반의 genealogy 그룹화 방법을 제안한다. 제안된 방법은 의미적 유사 관계와 퍼지 유사 관계를 이용하여 관계성을 결정하고 search(), insert(), hierarchy()의 연산을 이용하여 그룹화 구조를 수행한다. 이러한 구조는 그룹과 관련된 작업을 쉽게 처리해 주고, 질의응답, 객체 식별, 유사성 발견 등을 쉽게 해준다. 따라서 제안된 사서함 구조는 그룹화 생성을 통해서 사용자들에게 메시지 객체를 효율적으로 서비스하고 관리하는 기능을 제공한다. 그리고 제안된 기법의 성능을 알아보기 위해 5600개의 메시지 객체를 이용하여 non-grouping, BGM, RGM, OGM 방법과 비교 분석한다.

Abstract

Structuring mechanism, important to serve messages in post-office box structure, is to construct the hierarchy of classes according to the contents of message objects. This paper proposes α -cut based genealogy grouping method to cluster a lot of structured objects in application domain. The proposed method decides the relationship first by semantic similarity relation and fuzzy relation, and then performs the grouping by operations of search(), insert() and hierarchy(). This hierarchy structure makes it easy to process group-related processing tasks such as answering queries, discriminating objects, finding similarities among objects, etc. The proposed post-office box structure may be efficiently used to serve and manage message objects by the creation of groups. The proposed method is tested for 5600 message objects and compared with other methods such as non-grouping, BGM, RGM, OGM.

Key words : α -cut, message objects, genealogy grouping, post-office box

1. 서 론

최근에 IT 기반의 정보서비스 기술은 마이크로프로세서의 가격이 낮아지고 소형화 및 고성능화가 진행됨에 따라 제품 경쟁력의 핵심이 H/W 생산 기술에서 S/W 최적화 기술로 이동하는 변혁기를 맞이하여 임베디드 S/W가 탑재된 상품 가치가 우선시 되는 기술 집약적, 고부가가치 산업으로 발전하고 있다. 또한 많은 정보들이 디지털 형태로 관리되고 서비스됨에 따라, 텍스트, 오디오, 비디오 정보와 같은 멀티미디어 정보들을 관리하고 처리하기 위한 새로운 기법들이 제안되고 있다[2, 3, 4, 6, 11, 12]. 일반적으로 멀티미디어 정보 서비스에 있어서 미디어 타입은 색인과정과 검색 과정을 거

쳐 수행되며, 이때 질의어와 유사한 정보들이 검색되어 사용자들에게 서비스되게 된다. 이러한 유사한 정보들을 검색하고 서비스하기 위해서 의미적 유사도 척도(semantic similarity measure)[1, 5]가 이용되고 있으며, 이 기법은 정보 검색 및 정보 관리 분야에 있어서 중요한 역할을 수행하고 있다. 의미적 유사도는 동의어 집합, 의미적 이웃 객체들 간의 정합 과정을 함수적으로 수행해야 하는 문제가 발생되며, 동등한 개체 클래스들을 구성하는데 문제점이 발생되고 있다[9, 10, 11]. 본 논문은 임베디드 사서함 서비스를 위한 genealogy 그룹화 방법을 제안하며, 제안된 기법은 퍼지 필터링 기반의 α -cut 을 이용한 genealogy 그룹화 방법을 제안한다. 이때 그룹화를 수행하기 위하여 응용 메시지 정보를 프로파일 중심의 객체 타입으로 분류하며, 분류된 메시지 객체 정보들에 대해서 퍼지 유사관계 (Fuzzy Similarity Relation)를 수행하게 된다.

따라서 본 논문은 사용자가 언제 어디서나 시공간상의 제약 없이 받고 사용자의 인지 기능 없이 응용 메시지 정보를 프로파일 중심의 객체 타입으로 서비스를 받을 수 있도록

접수일자 : 2005년 1월 31일

완료일자 : 2005년 12월 5일

감사의 글 : 본 논문은 정보통신부 연구진흥원에서 지원하고 있는 정보통신 기초기술 연구지원사업의 연구 결과입니다.

α -cut 기반의 genealogy 그룹화 방법을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 관련 연구에 대해서 알아보고, 3장에서는 메시지 객체를 서비싱하기 위한 그룹화 모델에 대해서 살펴본다. 그리고 4장에서는 제안된 기법의 성능을 알아보기 위해 시뮬레이션 평가를 수행하고, 끝으로 결론 및 향후 연구방향에 대해서 살펴본다.

2. 관련 연구

퍼지 필터링 기반의 메시지 서비스를 위해서는 관련된 메시지 객체들을 유사도 관계에 따라 그룹화하는 방법이 요구되며, 이절에서는 이러한 방법을 위한 OGM, BGM, RGM 기법[9]에 대해서 살펴보기로 한다.

2.1 OGM

OGM(Optimistic Genealogy Measure)기법은 하나의 그룹의 각 원소들에 대해서 유사도를 계산하는 방법으로서 각 원소들에 대한 평균 가중치를 구하여 그룹들 간에 유사도를 계산하는 방법이다. 그룹 G1과 G2에 대한 임의의 자식 노드 k_i 에 대한 유사도는 결합성 집합과 노드 유사도를 이용하여 유사도 수행은 다음과 같다.

결합성 집합 : $Adhesion_{G_1, G_2}(k_i) = \{k_j \in G_2 \mid LCA(k_i, k_j) = LCA_{G_1, G_2}(k_i)\}$ 이다.

여기서 $Adhesion_{G_1, G_2}(k_i)$ 는 k_i 가 G_1, G_2 에서 결합되는 정도를 나타내는 결합성 집합

노드 유사도 : $leafsim_{(G_1, G_2)}(k_i) = depth(LCA_{(G_1, G_2)}(k_i) / depth(k_i)$ 이다.

여기서 $leafsim_{(G_1, G_2)}(k_i)$ 는 노드 k_i 가 그룹 G_1, G_2 에 있는 노드들과 얼마나 잘 결합되는지를 나타내는 유사도이다.

이 방법은 유사도를 이용한 문서 구조화 방법으로 최적의 기법을 제공하지만 평균 가중치를 구하여 문서의 일치성(matching)을 검사하여 결합성을 측정해야 하는 문제점이 발생되고 있다.

2.2 BGM

BGM(Balanced Genealogy Measure)기법은 OGM에서 발생하는 유사도 중복 문제를 해결하기 위한 방법으로서 [0, 1]사이의 매개변수(parameter) β 를 이용하여 결합성의 다양성(multiplicity)으로 인한 유사도가 떨어지는 문제를 해결하기 위한 기법이다. 그러나 이 기법은 [0,1]사이의 매개변수를 이용하여 결합성의 문제를 부분적으로 해결하였지만 같은 그룹에서 같은 타입을 가지고 있는 문서들에 대해서 결합성을 반복하여 측정해야 하는 문제점을 가지고 있다.

2.3 RGM

RGM(Recursive Genealogy Measure)기법은 다중 발생에 관한 문제를 해결하기 위한 기법으로서 하나의 트리 구조에서 많은 노드로 구성되어 있을 때 유사도의 중복성 수행으로 인한 문제점을 개선하기 위한 기법이다.

이 기법은 트리 구조 T1과 T2사이의 유사도를 자식노드일 때와

$$sim_{T1, T2}(n) = optleafsim_{T1, T2}(n),$$

내부 노드일 때로

$$sim_{T1, T2}(n) = \frac{\sum_{c \in CT1(n)} W_{T1}(c) \times sim_{T1, T2}(c)}{\sum_{c \in CT1(n)} W_{T1}(c)}$$

로 구분하여 유사도를 수행하게 된다.

이 방법은 객체 노드가 증가될 때에는 유사도의 중복 수행이 발생하는 문제점을 가지고 있다.

3. 그룹화 모델

본 논문에서 제안된 구조는 응용 도메인으로부터 추출된 메시지 객체를 프로파일 구조로 변환하여 이들의 관계성을 의미적 유사 관계(Semantic Similarity Relation)와 퍼지 유사 관계(Fuzzy Similarity Relationship)로 분류하여 수행한다. 그리고 이들과의 상호작용을 위한 시스템 구성도는 그림 1과 같다.

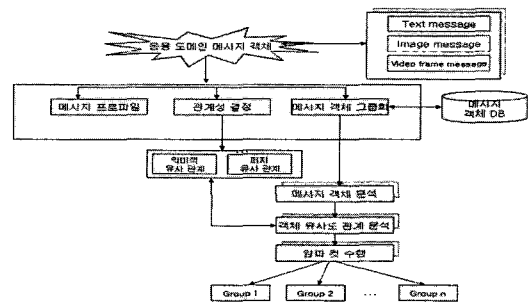


그림 1. 시스템 구성도
Fig 1. System framework

3.1 메시지 프로파일 구조

메시지 사서함 서비스 구조는 메시지 객체들을 프로파일 구조로 분류하여 객체들을 그룹화하게 되며, 이때 메시지 객체와 그룹들 간에는 관계성이 내포되며, 내포된 관계성에 따라 그룹화를 수행하게 된다.

응용 도메인에서 그룹화를 위한 메시지 객체 타입 ot (object type)는 $S = (ot, \beta)$ 로 표현되며, $S(\text{Set})$ 는 메시지 객체들의 집합체로서 응용 도메인에서 객체 $(ot_x, ot_y, ot_z) \in D_{\text{domain}}$ 의 조건은 다음과 같은 성질을 만족한다고 가정한다.

- (1) $p(ot_x, ot_y) = p(ot_y, ot_x)$
- (2) $0 \leq p(ot_x, ot_y) \leq 1, ot_x \neq ot_y$
- (3) $p(ot_x, ot_x) = 1$
- (4) $p(ot_x, ot_y) \leq p(ot_x, ot_z) + p(ot_z, ot_y)$

이와 같은 조건을 만족할 때 객체 관계성을 결정하기 위한 메시지 객체 프로파일의 구조는 그림 2와 같다.

ot	Profile-id				
	supergroup	subgroup	relationship	search	index
			SR	FR	

그림 2. 메시지 객체 프로파일 구조
Fig. 2 profile structure of message object

그림 2에서 객체 ot 는 메시지 객체이며, Profile-id는 메시지 객체를 식별하기 위한 식별자이다. supergroup과 subgroup 필드는 관계성을 기술하는 상위계층과 하위계층을 기

술하는 필드로서 상속성 계층 구조를 구성하게 된다. relationship 필드는 관계성을 설정하는 필드로서 의미적 유사 관계와 퍼지 유사 관계의 서브 필드로 구성된다. search 필드는 supergroup과 subgroup사이의 의미적 유사관계, 퍼지 유사 관계가 존재할 때 ot를 확장하기 위한 탐색필드이며, 노드 탐색을 위해 search()함수를 이용한다. 이때 상위 노드로의 탐색은 up-search()함수를 이용하며, 하위 노드로의 탐색은 down-search()함수를 이용한다. 그리고 index 필드는 그룹화를 수행할 색인 필드로서 true와 false로 구분하여 색인을 수행하게 된다. true는 색인과정에서 이미 참조한 메시지 객체가 존재할 때 색인을 중단하는 필드이고, false는 색인과정에서 참조되지 않은 메시지 객체를 색인하기 위한 필드이다.

예를 들어 음식도메인에서 추출된 ot “김치찌개”에 대한 프로파일 구조가 그림3과 같다고 가정하자.

김치 찌개	Profile-id				
	supergroup	subgroup	relationship	search	index
	한식	none	FR	up-search()	true

그림 3. 메시지 객체 프로파일 구조 예
Fig. 3 profile example of message object

그림 3에서 supergroup은 “한식”이고, subgroup “none”은 subgroup이 존재하지 않음을 의미한다. 그리고 relationship에서 퍼지유사도 관계(FR)은 의미적 유사도 관계(SR)에 의한 퍼지 관계가 유지됨을 의미하고 search는 메시지 객체 “김치찌개”가 상위노드로의 탐색이 수행됨을 의미한다.

index “true”는 메시지 객체가 의미적 유사도 관계와 퍼지 유사 관계를 위해 색인참조 되었음을 의미한다.

3.2 관계성 결정

메시지 사서함 구조에서 ot들은 supergroup과 subgroup 사이에 의미적 유사 관계와 퍼지 유사 관계를 가지게 되며, 또한 ot 와 ot, ot와 md 사이에 의미적 유사 관계와 퍼지 유사 관계를 가지게 된다. 본 논문에서 퍼지 유사 관계는 $0 \leq \mu \leq 1$ 의 값으로 표현되며, ot들 사이의 퍼지 유사 관계가 0.5 미만인 경우에는 관계성이 매우 적은 것으로 간주하여 $0.5 \leq \mu \leq 1$ 사이의 값들만 퍼지 유사 관계로 설정한다.

3.2.1 의미적 유사 관계

의미적 유사 관계는 클라이언트에서 요청한 ot구조를 파악하여 해당 ot의 생성, 삭제, 활성화, 비활성화와 같은 연산을 수행하게 되며, instance-of, whole-of, part-of, association, aggregation 등의 관계성을 이용하여 최상위 그룹의 ot가 가지고 있는 속성들을 하위 ot에게 상속하거나 일부만을 상속하는 기능을 수행하게 된다. 이러한 관계가 성립될 때 ot와 ot, ot와 md간의 의미적 관계성은 다음과 같이 정의된다.

(정의 1) ot와 ot들 간의 의미적 관계성(SR)은 $SR(ot) = \{md | md \in MD \text{ and } \langle md, ot \rangle \in \text{의미적 유사 관계}(SR)\}$ 이며, 여기서 SR(ot)는 ot들 간의 의미적 유사관계가 성립되는 의미적 관계성이다.

3.2.2 퍼지 유사 관계

퍼지 유사 관계는 의미적 유사 관계를 구성하는 ot들을 참

조하여 그룹화를 활성화하게 되며, 의미적 유사 관계가 수행되지 않는 ot에 대해서는 퍼지 필터링을 수행하게 된다. 따라서 본 논문에서는 퍼지 유사도 관계성을 위해 α -cut관계, 유사관계, 호환관계에 의해 관계성을 결정하게 된다.

(1) α -cut 관계

α -cut 관계[7, 8]은 [0,1]사이의 퍼지 값에서 임의의 $\alpha(0 \leq \alpha \leq 1)$ 이 되는 구간을 나타내며, 퍼지 집합의 원소들에 대한 소속 기준을 정의할 때 사용된다. α -cut관계에 의한 퍼지 유사도 관계성은 $S=(ot, p)$ 에 대하여 $\mu_s:ot \rightarrow [0,1]$ 로 정의되며, μ_s 는 유사관계에 의해서 퍼지 유사도 관계성이 부여되는 membership function이다. 만일 md에서 임의의 ot를 원소로 하는 퍼지 집합 ot에 대해서 $\alpha \in [0,1]$ 이라면 ot에 대한 퍼지 집합 ot와 α -cut은 다음과 같이 정의된다.

(정의 2) $ot \alpha=(ot | md(ot) \geq \mu)$ 이다.

여기서 $ot \alpha$ 는 ot들에 대한 퍼지집합이 μ 값 이상으로 이루어진 집합을 의미한다.

(정의 3) α -cut= $(ot | md(ot) \geq \mu)$ 이다.

α -cut은 퍼지 유사도 관계성을 [0,1]사이의 값에서 임의의 $\mu(0 \leq \mu \leq 1)$ 값이 소속되는 정도를 퍼지집합으로 표현한 값이다.

예를 들어 $md(ot) \geq \mu$ 를 구성하는 md와 ot사이의 퍼지관계가 표1과 같다고 가정하자.

표 1. md와 ot사이의 퍼지관계
Table. 1 Fuzzy relation between md and ot

	MD	ot ₁	ot ₂	ot ₃	ot ₄	ot ₅	ot ₆	ot ₇	ot ₈	ot ₉	ot ₁₀
OT											
	md1	0.43	0.87	0.32	0.97	0.75	0.12	0.06	0.49	0.54	0.65

표 1에서 $md(ot) \geq 0.5$ 를 만족하는 0.5-cut은 {ot₂, ot₄, ot₅, ot₉, ot₁₀}이 되며, $md(ot) \geq 0.5$ 를 만족하지 않는 ot₁, ot₃, ot₆, ot₇, ot₈은 필터링이 수행된다. 이와 같이 필터링된 객체들을 퍼지 필터링 객체라 하며, 퍼지 필터링에 의해 퍼지 유사도 관계가 성립되게 된다.

(2) 유사관계

유사관계는 MD에서 추출한 ot들에 대해서 퍼지 유사도 관계성을 수행하는 관계성으로서 이들은 의미적으로 내포하는 관계성이 서로 다르다. 또한 퍼지 유사도 관계를 구성하는 ot와 ot, ot와 md사이에는 퍼지 유사 관계가 서로 다르며, 이때 퍼지 유사 관계를 만족하는 ot들은 유사 클래스가 생성되게 된다. 이와 같은 방법은 ot와 ot, ot와 md사이의 관련 정도를 파악하여 관련성이 낮은 ot들을 필터링하기 위한 방법이다. ot와 ot, ot와 md사이의 관련 정도를 파악하기 위한 관계성은 반사관계, 대칭관계, 전이관계를 수행하여 파악되며, 반사관계, 대칭관계, 전이관계는 다음과 같다.

- 1) 반사관계 : $\mu \cong (ot_x, ot_x)=1$
- 2) 대칭관계 : $\mu \cong (ot_x, ot_y) = \mu \cong (ot_y, ot_x)$
- 3) 전이관계 : $\mu \cong (ot_y, ot_z) \geq \min\{\mu \cong (ot_x, ot_y), \mu \cong (ot_y, ot_x)\}$

이와 같은 관계가 성립될 때 ot들 사이의 퍼지 유사관계는 다음과 같이 정의된다.

(정의4) $\mu_{md1=md2} = \max\{\min\{\mu_{md1}(ot), \mu_{md2}(ot)\}, \min\{1 - \mu$

$\mu_{md1}(ot), 1 - \mu_{md2}(ot))$ 이다.

여기서 $\mu_{md1}(ot)$ 는 md_1 에서 임의의 ot 에 대한 퍼지 유사정도이며, $\mu_{md2}(ot)$ 는 md_2 에서 임의의 ot 에 대한 퍼지 유사정도이다.

또한 ot 와 ot 들 사이의 퍼지 유사관계는 각 md 를 구성하고 있는 ot 들의 퍼지 유사 정도에 의해 표현이 되며, md 에서의 ot 와 ot 들에 대한 평균 퍼지 유사 관계는 다음과 같이 정의된다.

(정의5) $\mu_{ij} = \mu_{wi \equiv wj} = (1/N) \sum_{n=1}^m \mu_{wi \equiv wj} (md_n)$ 이다.

여기서 μ_{ij} 는 oti 와 otj 사이의 유사정도를 의미하며, N 은 전체 md 의 수를 의미한다. 그리고 $\mu_{wi \equiv wj}$ 는 임의의 md 에서 oti 와 otj 사이의 퍼지 유사정도를 의미한다.

예를 들어 md_1 과 md_2 에서의 (정의4)와 (정의5)를 이용한 ot 의 퍼지 유사 정도가 표2와 같다고 가정하자.

표 2. 퍼지 유사정도 ($\mu_{wi \equiv wj}$)
Table 2. Fuzzy similarity ($\mu_{wi \equiv wj}$)

	md ₁							
md ₂		ot ₁	ot ₂	ot ₃	ot ₄	ot ₅	ot ₆	ot ₇
ot ₁		1	0.32	0.78	0.45	0.84	0.55	0.92
ot ₂		0.32	1	0.57	0.68	0.95	0.47	0.26
ot ₃		0.78	0.57	1	0.62	0.35	0.48	0.81
ot ₄		0.45	0.68	0.62	1	0.52	0.24	0.85
ot ₅		0.84	0.95	0.35	0.52	1	0.66	0.02
ot ₆		0.55	0.47	0.48	0.24	0.66	1	0.72
ot ₇		0.92	0.26	0.81	0.85	0.02	0.72	1

표 2.에서처럼 α -cut을 이용하여 퍼지 유사정도를 만족하는 ot 들의 유사관계 클래스 생성이 가능하며, md_2 에서 ot_1 에 대한 퍼지 유사 정도를 0.5 -cut로 적용한 유사관계는 0.5 -cut={ot₁, ot₃, ot₅, ot₆, ot₇}, ot_2 에 대한 유사관계는 0.5 -cut={ot₂, ot₃, ot₄, ot₅}, ot_3 에 대한 유사관계는 0.5 -cut={ot₁, ot₂, ot₃, ot₄, ot₇}, ot_4 에 대한 유사관계는 0.5 -cut={ot₂, ot₃, ot₄, ot₅, ot₇}, ot_5 에 대한 유사관계는 0.5 -cut={ot₁, ot₂, ot₄, ot₅, ot₆}, ot_6 에 대한 유사관계는 0.5 -cut={ot₁, ot₅, ot₆, ot₇}이고, ot_7 에 대한 유사관계는 0.5 -cut={ot₁, ot₃, ot₄, ot₆, ot₇}이 된다. 따라서 반사관계, 대칭관계, 전이관계를 만족하는 임의의 ot 들은 퍼지 유사정도에 대한 관련정도 값을 부여받고 α -cut에 따라 관련 ot 들을 생성하여 그룹화를 수행하게 된다.

(3) 호환관계

호환관계는 md 에서의 ot 발생빈도를 이용하며, 발생빈도가 많을수록 퍼지 관계 유사도가 높다고 할 수 있다. 호환관계는 반사관계와 대칭관계에 의해서 퍼지 유사관계가 결정되며, α -cut={ot | md(ot) ≥ μ}을 만족하는 ot 와 md 들을 호환관계에 의한 퍼지 유사 정도로 간주한다. 본 논문에서는 호환관계에 의한 퍼지 유사도를 위해서 FAS(Fuzzy Average Similarity)[15]를 이용하며, FAS는 해당 ot 가 임의의 md 와 얼마나 잘 일치되는지를 결정하는 퍼지 유사도 관계이다. 이 기법은 하나의 md 에서 많은 ot 들로 구성된 유사도 중복성 문제를 해결하기 위해 사용되며, 다음과 같이 정의된다.

(정의 6) FAS = $\sum_{G_i \text{는 부모}(G_i) \text{의 자식}} (\alpha\text{-cut}=\{ot \mid md(ot) \geq \mu\})$

/k이다.

여기서 k는 md 를 구성하는 ot 들의 전체 수이다.

예를 들어 ot 와 md 들의 퍼지 유사 관계가 표3과 같다고 가정하자.

표 3. ot 와 md 들 사이의 퍼지 유사 관계
Table 3. Fuzzy similarity relation between ot and md

	md						
ot		md ₁	md ₂	md ₃	md ₄	md ₅	md ₆
ot ₁		0.75	0.53	0.32	0.31	0.54	0.84
ot ₂		0.21	0.65	0.40	0.96	0.49	0.79
ot ₃		0.06	0.76	0.61	0.57	0.60	0.92
FAS		0.34	0.65	0.44	0.61	0.54	0.85

표 3. 에서 보듯이 α -cut={ot | md(ot) ≥ μ}를 0.6 -cut으로 적용할 때의 FAS ≥ 0.6을 만족하는 퍼지 유사 관계는 {md₂, md₄, md₆}가 된다. 이는 FAS를 만족하는 ot 들이 md 상에 반복적으로 존재함에 따라 유사관계가 높은 ot 들의 그룹화가 가능함을 의미하게 된다.

3.3 객체 그룹화 구성

그룹화는 메시지 사서함 구조에서 ot 들을 관리하고 서비스하기 위한 분류구조로서 계층적인 그룹화 구조를 가지게 된다. 객체 그룹화를 구성하기 위한 연산 구조는 노드 확장 및 탐색을 위한 search() 연산, 퍼지 유사 관계에 의해 관련된 ot 를 삽입하기 위한 insert() 연산 그리고 계층 구조를 구성하는 hierarchy() 연산 구조로 구성된다.

3.3.1 search()에 의한 그룹화

search() 연산은 퍼지 유사도 관계 조건을 만족하는 ot 들에 대해서 α -cut={ot | md(ot) ≥ μ}을 수행하여 그룹에 추가시키는 연산으로서 insert() 연산이 수행되는 도중에 수행된다. search() 연산을 수행하는 도중에 index 연산을 수행하여 ot 들에 대한 중복성 문제를 피하게 되며, up-search()와 down-search()를 이용하여 supergroup과 subgroup으로 탐색을 수행하게 된다. 이처럼 ot 들에 대한 search() 연산을 수행하는 과정은 다음과 같다.

```

Search()_ot
ot : search()를 수행하기 위한 ot1, ot2,..., otn
begin
if 의미적 유사관계와 퍼지 유사도 관계가 존재하는 ot가 md에 존재
// { α-cut={ot | md(ot) ≥ μ}를 수행한다.
if 0.5-cut={ot | md(ot) ≥ 0.5}이 존재
then ot를 group에 삽입
else if (up_search()
and down_search())
then begin
0.5-cut={ot | md(ot) ≥ 0.5}를 수행한다.
end
ot들에 대한 group을 갱신한다.
end
end
    
```

예를 들어 그림 4는 supergroup "점심"과 md1에서 추출된 4개의 ot "자장면", "돈까스", "커피", "한식"이 구성되어 있다고 가정할 때 Search()_ot를 수행한 과정이다.

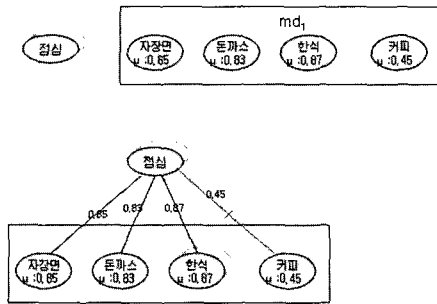


그림 4. search() 연산
Fig. 4. search() operation

그림 4의 md1에서 0.5-cut={ot | md(ot) ≥ 0.5}를 만족하는 ot들은 "자장면", "돈까스", "한식"이고, 이들이 supergroup "점심"과 의미적 유사도 관계, 퍼지 유사도 관계로 구성되어 있을 경우 md1의 ot들에 대한 index를 수행한 후 관계성에 따른 그룹이 구성되게 된다. 따라서 search()연산과 index를 수행한 후 "자장면", "돈까스", "한식"은 0.5-cut을 만족하는 ot이고, "커피"는 그렇지 않은 ot이므로 퍼지 유사도 관계 관계를 수행하여 md(ot) ≥ 0.5을 만족하는 ot와 그룹화를 수행하게 된다.

3.3.2 insert()에 의한 그룹화

insert()에 의한 그룹화는 관련 group에 md(ot) ≥ 0.5를 만족하는 새로운 ot를 삽입하는 연산으로서 삽입 연산 수행 중에 search()와 index를 수행하여 그룹화를 수행하게 된다. 새로운 ot가 md에 존재할 때 insert()에 의한 삽입과정은 다음과 같다.

```

Insert_md(ot)
// md(ot) : md에서 insert를 수행할 ot들이 존재
ot : insert를 수행하기 위한 ot1, ot2, ..., ot_n
퍼지 유사도 관계 : md(ot) ≥ 0.5을 만족하는 ot들의 퍼지 관계
begin
  Search()_ot 수행
  if( md(ot) ≥ 0.5을 만족하는 group이 존재)
    group에 퍼지 유사도 관계를 만족하는 ot 삽입
    hierarchy() 수행
  else md(ot) ≥ 0.5을 만족하지 않는 ot에 대해서 filtering 수행.
end
    
```

예를 들어 그림 5는 $R \subseteq md \times ot$ 에서 $md_1 = \{(김밥, 0.75), (녹차, 0.36)\}$, $md_2 = \{(김치찌개, 0.82), (스파게티, 0.24)\}$, $md_3 = \{(우동, 0.48), (닭집, 0.55)\}$ 라고 가정할 때 Insert()_ot(md)를 수행한 과정이다.

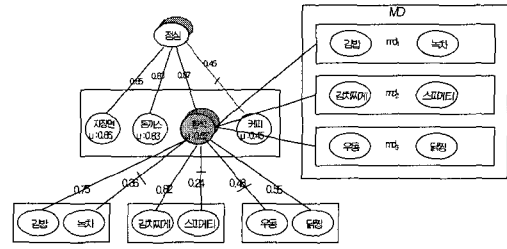


그림 5. insert() 연산
Fig. 5. insert() operation

그림 5에서 보듯이 $md_1 = \{(김밥, 0.75), (녹차, 0.36)\}$, $md_2 = \{(김치찌개, 0.82), (스파게티, 0.24)\}$, $md_3 = \{(우동, 0.48), (닭집, 0.55)\}$ 을 탐색한 후 $md(ot) \geq 0.5$ 를 결정하게 된다. md_1 에서 $md(ot) \geq 0.5$ 을 만족하는 ot는 "김밥", md_2 에서 $md(ot) \geq 0.5$ 을 만족하는 ot는 "김치찌개", md_3 에서 $md(ot) \geq 0.5$ 을 만족하는 ot는 "닭집"이므로 이들과 퍼지 유사도 관계 수행하여 삽입을 수행한다. 이때 필터링을 수행하여 $md(ot) \geq 0.5$ 를 만족하지 않는 ot들은 필터링되며 $md(ot) \geq 0.5$ 를 만족하는 "김밥", "김치찌개", "닭집"과 supergroup "한식"과의 퍼지 유사도 관계 의해 그룹화가 수행된다.

3.3.3 hierarchy()에 의한 그룹화

hierarchy()에 의한 그룹화는 그룹에 ot를 삽입할 때 supergroup과 subgroup에 관한 그룹을 탐색하여 관련 ot들을 점차적으로 그룹화 하는 방법이다. 이때 퍼지 유사도 관계는 ot들을 점차적으로 탐색하여 관계성을 결정하게 되며, 관련된 ot들의 관련성을 탐색하기 위하여 search()를 수행한다. 퍼지 유사도 관계 가진 ot가 계층구조에 삽입될 때 해당 ot의 super가 계층구조에 존재하는지 그렇지 않은지를 탐색하여 supergroup과 퍼지 유사도 관계를 수행하거나 새로운 group을 생성하게 된다. hierarchy()을 수행하는 과정은 다음과 같다.

```

Make_hierarchy(ot)
//ot : hierarchy()를 수행하는 ot1, ot2, ..., ot_n
begin
  while (모든 ot들에 대해서) do
    search() 수행
    if (ot.supergroup이 존재)
      begin
        <supergroup, ot, 퍼지 유사도 관계>
        and <supergroup, subgroup, 퍼지 유사도 관계>수행
        if((ot.supergroup.supergroup)
           이 존재)
          then
            ot.supergroup.supergroup의 subgroup에 ot.supergroup 기록:
          end
        else begin
          hierarchy 갱신;
        end
      end
    end
  end
end
    
```

예를 들어 그림5에서 삽입될 ot 구조가 그림6과 같을 때 hierarchy()를 수행하기 위해 supergroup이 존재하는지를 탐색하게 된다. 그림6에서 "후식"은 ot이자 "커피", "쥬스", "과

일"을 sub로 갖는 supergroup이며, "과일"은 "사과", "딸기"를 sub로 갖는 supergroup이자 ot가 된다. 따라서 <supergroup, ot, 퍼지유사도관계> 와 <supergroup, subgroup, 퍼지유사도관계>를 수행하여 퍼지유사도를 결정하며, 이들 관계를 수행 한 후의 계층 구조는 그림7과 같다.

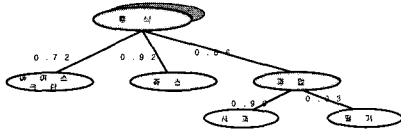


그림 6. 삽입될 ot 구조

Fig. 6. ot structure which will be inserted

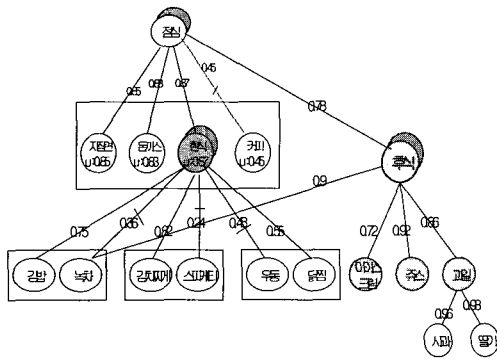


그림 7. hierarchy() 연산

Fig. 7. hierarchy() operation

그림 7에서 직 간접적으로 연결되어 있지 않은 ot들 사이의 퍼지유사도 관계 정의되지 않으며 예를 들어 "후식"과 "커피"사이에는 관련정도는 존재하지만 예지가 존재하지 않기 때문에 퍼지유사도 관계에 의한 관련정도는 정의되지 않는다. hierarchy() 연산에서는 간접적으로 연관되는 ot들 사이의 합성 소속성 퍼지유사도 관계에 대해 퍼지 정형화 관계를 수행하며, 퍼지 정형화 관계는 다음과 같이 정의된다.

(정의7) 모든 ot에 대하여 $\mu_{FR}(ot) = \max(\min(\mu_{FRi}(ot), \alpha))$ 이다.

예를 들어 super "점심"과 sub "녹차" 사이의 퍼지 정형화 관계는 $\mu_{점심}(녹차) = \max(\min(\mu_{한식}(녹차), \mu_{점심}(한식)), \min(\mu_{후식}(녹차), \mu_{점심}(후식)))$ 즉 $\max(\min(0.36, 0.87), \min(0.90, 0.78)) = 0.78$ 이 된다. 이 예에서 두 경로가 모두 ot "녹차"를 경유하지만 ($\mu_{한식}(녹차), \mu_{점심}(한식)$)의 경로보다는 ($\min(\mu_{후식}(녹차), \mu_{점심}(후식))$)의 경로가 보다 정형화됨을 의미하게 된다.

4. 시뮬레이션 평가

본 논문에서 제안된 α -cut 기반의 genealogy 그룹화의 성능을 평가하기 위하여 네이버, 엠파스, 다음, 야후 코리아 등의 검색 엔진 사이트에 등록된 음식, 요리, 식품과 관련된 웹문서, 이미지, 지식, 사전, 뉴스, 전문자료, 멀티미디어 등의 데이터와 생활요리, 도시락, 간식의 텍스트를 실험 평가 자료로 이용하였다. 실험 측정을 위해서 서버는 Windows-2000

서버 상에서 SQL 2000 server를 이용하였으며, 클라이언트는 windows-xp의 Microsoft Visual C++6.0과 MFC를 이용하여 시뮬레이션 하였다. 시뮬레이션 데이터는 음식, 요리, 식품과 관련된 카테고리에서 텍스트, 이미지, 프레임의 5600개 OT정보를 G1, G2, G3, G4, G5의 5개의 그룹으로 분류하여 생성하였으며, 각각의 생성된 그룹에 대하여 α -cut을 0.6-cut, 0.7-cut, 0.9-cut으로 하여 시뮬레이션을 수행하였다. 시뮬레이션 수행 방법은 본 논문에서 제안된 방법과 non-grouping, BGM, RGM, OGM 방법으로 나누어 실행하며, 성능평가를 위한 척도로는 정확률과 평균 응답률을 이용하여 평가를 수행하였다. 정확률이란 MD중에서 그룹화가 수행된 OT의 비율을 말하며, 평균 응답률이란 질의를 수행한 후의 반응속도를 의미한다.

$$\text{정확률} = \frac{\text{그룹화가수행된OT}}{\text{MD중에서그룹화를수행하기위한OT}}$$

$$\text{평균응답률} = \text{OT}_{\text{time}}(1+1/N) + \frac{\text{OT}}{\text{MD}} \times \text{OT}_{\text{searchtime}}$$

여기서 OT_{time} 은 α -cut $\geq \mu$ 를 만족하는 OT를 검색하는 시간이며, $\text{OT}_{\text{searchtime}}$ 은 해당 OT를 탐색하는데 걸리는 시간이다.

다음은 본 논문에서 제안된 방법, non-grouping, BGM, RGM, OGM 방법에 대하여 α -cut을 0.6-cut, 0.7-cut, 0.9-cut이상으로 하여 각 그룹에 대한 정확률과 평균 정확률을 수행한 시뮬레이션 결과이다.

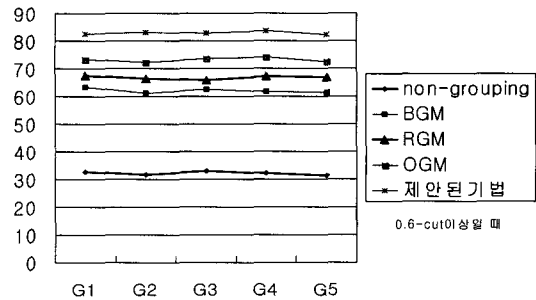


그림 8. 0.6-cut 이상일 때 정확률

Fig.8. Precision over 0.6-cut

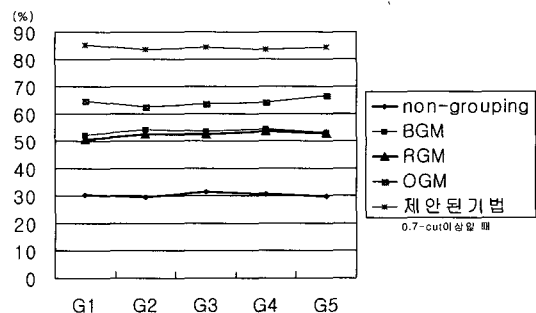


그림 9. 0.7-cut 이상일 때 정확률

Fig. 9. Fig.8. Precision over 0.7-cut

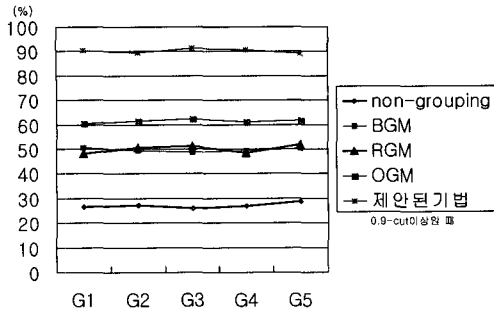


그림 10. 0.9-cut 이상일 때 정확률
Fig. 10. Precision over 0.9-cut

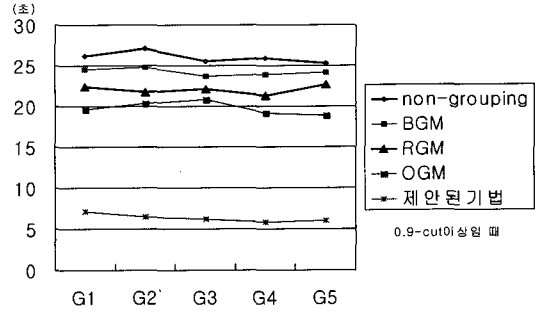


그림 13. 0.9-cut 이상일 때 평균응답률
Fig. 13. Average response ratio over 0.9-cut

표 4. α -cut에 따른 평균 정확률
Table 4. Average precision by α -cut

방법	non-grouping	BGM	RGM	OGM	제안된 기법
0.6-cut 이상일 때	32.2	62.04	66.8	73.2	82.9
0.7-cut 이상일 때	30.3	53.4	52.4	64.3	84.3
0.9-cut 이상일 때	27.02	49.7	50.1	61.4	90.3

그림과 표 4에서 보듯이 0.6-cut 이상일 때 제안된 기법의 평균 정확률은 82.9이며, non-grouping은 32.2, BGM은 62.04, RGM은 66.8, OGM은 73.2임을 알 수 있다. 즉 genealogy를 이용한 기법에서 비교적 효율적인 기법이라고 하는 OGM보다 0.6-cut 이상일 때 약 20%, 0.7-cut 이상일 때 약 20%, 그리고 0.9-cut 이상일 때 약 30% 정도의 성능이 향상됨을 알 수 있다.

그림 11, 12, 13은 각 그룹에 대하여 5회의 질의를 수행한 후의 평균응답률이며, 그림에서 본 바와 같이 제안된 방법이 non-grouping, BGM, RGM, OGM 방법에 비하여 해당 OT를 탐색하는데 걸리는 평균 응답률이 적게 걸리게 됨을 알 수 있다. non-grouping, BGM, RGM, OGM 방법은 임의의 문서 객체를 검색엔진 상에서 키워드를 이용하여 해당 객체를 탐색하는데 반해서 제안된 기법은 해당 객체를 α -cut $\geq \mu$ 를 수행하여 SR관계와 FR관계를 만족하는 OT를 그룹 안에서 탐색하기 때문이다. 그림에서 보는 것처럼 non-grouping, BGM, RGM, OGM 방법은 시간적 총합이 α -cut $\geq \mu$ 값이 증가함에 따라 평균 응답률이 증가되고 있지만 제안된 방법은 평균응답률이 줄어들었음을 알 수 있다.

따라서 제안된 방법은 관련된 메시지 객체 그룹화에 있어서 검색성능과 서비스율이 우수함을 알 수 있으며, 메시지 사서함 구조에 대해서 메시지 접근 및 문서 정보의 접근을 용이하게 지원해 줄 수 있는 장점을 가지게 된다.

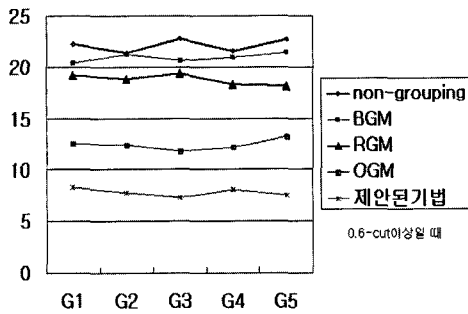


그림 11. 0.6-cut 이상일 때 평균응답률
Fig. 11. Average response ratio over 0.6-cut

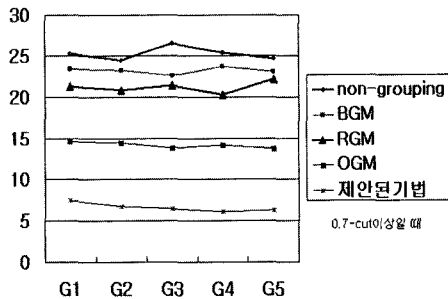


그림 12. 0.7-cut 이상일 때 평균응답률
Fig. 12. Average response ratio over 0.7-cut

5. 결론 및 향후 연구

최근에 임베디드에 대한 관심이 높아지면서 임베디드 기반 기법을 여러 산업 분야에 적용하기 위한 여러 가지 기법들이 연구 되고 있다. 따라서 본 논문에서는 무선망 및 특정 응용 도메인에서 서비스 요구가 발생될 때 서비스 접근 속도를 증가시켜주기 위한 퍼지필터링 기반의 genealogy 그룹화 기법을 제안하였다. 특히 메시지 객체들을 그룹화하기 위하여 OT들을 프로파일 타입으로 설정하여 SR관계와 FR관계에 의해 관계성이 결정되도록 하였으며, 관계성 결정을 위해서 α -cut $\geq \mu$ 를 이용하였다. genealogy 그룹화는 α -cut $\geq \mu$ 에 의해 수행되며 그룹화는 search()에 의한 그룹화, insert()에 의한 그룹화, hierarchy()에 의한 그룹화 방법을 제안하였다. 그리고 제안된 방법의 성능을 알아보기 위하여 음식 도메인에서 5600개의 OT를 대상으로 non-grouping, BGM, RGM, OGM 방법과 비교하여 정확률과 평균 응답률을 시뮬레이션 하였다. 그 결과 제안된 방법이 non-grouping, BGM, RGM, OGM 방법에 비해서 성능이 효율적임을 알 수 있었다. 향후 연구를 위해서는 제안된 방법이 서비스가 수행될 수 있도록 임베디드 툴 키트를 이용한 임베디드 시스템 개발이 연구되어야 할 것으로 사료된다.

참고 문헌

[1] Andrea Rodriguez, and Max J. Egenhofer, "Determining Semantic Similarity among Entity Classes from Different Ontologies," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 15, NO. 2, 2003, pp.442-456.

[2] Elisa Bertino, Jianping Fan, Elena Ferrari, Mohand-Said Hacid, Ahmed K. Elmagarmid, and Xingquan Zhu, "A Hierarchical Access Control Model for Video Database Systems," ACM Transactions on Information Systems, vol. 21, 2003, pp. 155-191.

[3] Elina Megalou, and Thanasis Hadzilacos, "Semantic Abstractions in the Multimedia Domain," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 15, NO. 1, 2003, 136-160.

[4] Filip Perich, Anupam Joshi, Senior Member, IEEE, Timothy Finin, and Yelena Yesha, Senior Member, IEEE, "On Data Management in Pervasive Computing Environment," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 5, 2004, PP. 621-633.

[5] Giuseppe Amato, Fausto Rabitti, Pasquale Savino, and Pavel Zezula, "Region proximity in Metric Spaces and its use for approximate similarity search," ACM transactions on Information Systems, vol. 21, No. 2, 2003, pp. 192-227.

[6] Guojun Lu, "Techniques and Data Structure for Efficient Multimedia Retrieval Based on Similarity," IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 4, NO. 3, 2002, pp. 372-384.

[7] Jung-Hsien Chiang, Shihong Yue, and Zong-Xian Yin, "A new fuzzy cover approach to clustering," IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 12, NO. 2, 2004, PP. 199-208.

[8] Laszlo T. Koczy and T. Geodeon, "Information Retrieval by Fuzzy Relations and Hierarchical Co-occurrence," Part I. TR97-01, Dept. of Info. Eng., School of Com. Sci. & Eng., UNSW, 1997, pp. 1-18.

[9] Prasanna Ganesan, Hector Garcia-Molina, and Jennifer Widom, "Exploiting Hierarchical Domain Structure to Compute Similarity," ACM Transactions on Information Systems, vol. 21, 2003, pp. 64-93.

[10] Stuart E. Middleton, Nigel R. Shadbolt, and David C. DE Roure, "Ontological User profiling in Recommender Systems," ACM Transactions on Information systems, Vol. 22, No. 1, 2004, pp. 54-58.

[11] Thomas Hofmann, "Latent Semantic Models for Collaborative Filtering," ACM Transactions on Information systems, Vol. 22, No. 1, 2004, pp. 89-115.

[12] 이종득, 김대경, " 퍼지 필터링 구조를 이용한 멀티미디어 통계 사서함 시스템," 정보처리학회논문지 제11-Brnjs 제6호, 2004.10, pp. 709-716. 제11권 제6호, 2004.11, pp. 86-99.

저자 소개



이종득(Chong-Deuk Lee)

1983년 2월 : 전북대학교 전산통계학과 졸업(이학사)
 1989년 2월 : 전북대학교 대학원 전산통계학과 졸업(이학석사)
 1998년 2월 : 전북대학교 대학원 전산통계학과 졸업(이학박사)
 1992년 3월~2002년 2월 : 서남대학교 컴퓨터정보통신학과 교수
 2002년 3월~2005년 현재 : 국립 익산대학교 정보통신공학과 교수

관심분야 : 멀티미디어 시스템, 멀티미디어 통신, 무선인터넷, 무선통신 등
 E-Mail : cdlee1008@iksan.ac.kr



안정용(Jeong-Yong Ahn)

1992년 2월 : 전북대학교 전산통계 졸업
 1994년 2월 : 전북대학교 전산통계학과 졸업(이학석사)
 2001년 8월 : 전북대학교 대학원 전산통계학과 졸업(이학박사)
 1995년 3월~2003년 8월 : 서남대학교 컴퓨터정보통신학과 교수
 2003년 8월~현재 : 전북대학교 통계정보학부 교수

관심분야 : 데이터 마이닝, 멀티미디어 시스템, 통계데이터베이스, 무선인터넷 등
 E-Mail : jyahn@chonbuk.ac.kr