

질의 응답 시스템에서 지식 설명의 의미적 포함 관계를 고려한 의미적 퍼지 함의 연산자

Semantic Fuzzy Implication Operator for Semantic Implication Relationship
of Knowledge Descriptions in Question Answering System

안찬민*, 이주홍*, 최범기*, 박선**

인하대학교 IT공과대학 컴퓨터정보공학부*, 목포대학교 MTRC 센터 정보산업연구소**

Chan-Min Ahn(ahnch1@datamining.inha.ac.kr)*, Ju-Hong Lee(juhong@inha.ac.kr)*,
Bumghi Choi(bumghichoi@yahoo.co.kr)*, Sun Park(sunpark@mokpo.ac.kr)**

요약

질의 응답 시스템은 사용자의 질의에 대해 다른 사용자의 응답을 저장하고 보여 주는 시스템이다. 사용자의 질의를 만족시키는 응답을 정확히 검색하고자 노력하는 많은 연구들이 있었지만 이에는 근본적인 한계가 있었다. 따라서 질의 응답 시스템에서는 보조적인 방법으로 사용자의 질의를 만족시킬 가능성이 높은 다른 질의를 추천하는 방법이 사용되고 있다. 이전 연구에서 내용적으로 포함하는 정도가 큰 질의를 하위 질의로서 추천하는 내용 기반 추천 방법으로서 퍼지 관계 곱 연산자(fuzzy relational product operator)를 사용하는 방법이 제안되었고, 기본적인 함의 연산자로서 Kleene-Dienes 연산자가 사용되었다. 하지만 Kleene-Dienes 연산자는 설명의 의미적 포함관계를 고려한 방법이 아니기 때문에 질의응답의 의미적 포함 정도를 계산하기에 적합하지 않다. 본 논문에서는 두 질의에 대한 설명의 의미적 포함관계를 고려한 새로운 함의 연산자를 제안한다. 새로운 연산자는 어떤 질의 및 응답 들이 다른 질의와 그 응답들에 의미적으로 포함되는 정도를 계산하도록 설계되었다. 실험을 통하여 새로운 함의 연산자를 적용한 퍼지 관계곱 연산자를 사용하면 사용자가 원하는 지식을 추천할 가능성이 높아짐을 보였다.

■ 중심어 : | 질의 응답 시스템 | 퍼지 관계 곱 연산자 | 의미적 퍼지 함의 연산자 | 질의 추천 |

Abstract

The question answering system shows the answers that are input by other users for user's question. In spite of many researches to try to enhance the satisfaction level of answers for user question, there is a essential limitation. So, the question answering system provides users with the method of recommendation of another questions that can satisfy user's intention with high probability as an auxiliary function. The method using the fuzzy relational product operator was proposed for recommending the questions that can includes largely the contents of the user's question. The fuzzy relational product operator is composed of the Kleene-Dienes operator to measure the implication degree by contents between two questions. However, Kleene-Dienes operator is not fit to be the right operator for finding a question answers pair that semantically includes a user question, because it was not designed for the purpose of finding the degree of semantic inclusion between two documents. We present a novel fuzzy implication operator that is designed for the purpose of finding question answer pairs by considering implication relation. The new operator calculates a degree that the question semantically implies the other question. We show the experimental results that the probability that users are satisfied with the searched results is increased when the proposed operator is used for recommending of question answering system.

■ keyword : | Question Answering System | Fuzzy Relational Product Operator | Semantic Fuzzy Implication Operator | Question Recommendation |

* 본 논문은 인하대학교의 지원에 의하여 연구되었음.

접수번호 : #110121-001

접수일자 : 2011년 01월 21일

심사완료일 : 2011년 03월 16일

교신저자 : 안찬민, e-mail : ahnch1@datamining.inha.ac.kr

I. 서론

질의 응답 시스템은 사용자가 원하는 정보를 얻기 위해 질문을 하면 그에 대한 응답을 다른 사람들이 제공할 수 있도록 하는 시스템이다. 질의응답은 어떤 사용자가 입력한 질의와 그에 대해 다른 사용자가 응답한 내용으로 이루어진다. 이러한 질의응답은 지식이라고 표현하기도 한다. 질의 응답 시스템에 저장된 지식들은 매우 정확한 정보를 가질 확률이 높다. 왜냐하면 지식의 내용은 질의에 대해 다른 사용자나 전문가가 직접 응답한 내용이기 때문이다.

그래서 현재 많은 포털 서비스 업체는 질의 응답 서비스를 제공하고 있다. 영어를 사용하는 대표적인 질의 응답 시스템에는 Yahoo! Answers, Answer.com, askville 등이 있고 한글을 사용하는 질의 응답 시스템에는 네이버 지식-IN, Daum 지식,네이트 지식 등이 있다. 질의 응답 시스템은 비교적 짧은 기간에도 불구하고 이용자의 수가 증가함에 따라 최근 질의 응답 시스템에 관한 많은 연구가 이루어지고 있다[2][3][6].

그러나 이러한 연구들에도 불구하고 시스템이 사용자가 만족할 수 있는 검색 결과를 찾아주는 데에는 한계가 있다. 심지어 시스템은 질의에 부합하는 응답이 있음에도 그 응답을 찾지 못할 수도 있다. 이는 시스템이 사용자의 의도를 정확하게 이해하기 어렵기 때문이다. 그러나 사용자의 질의 의도를 시스템에 명확하게 전달하는 것은 어렵다. 이는 사용자가 입력한 질의의 표현이 부족하거나 사용된 단어의 다의성 등으로 질의에 사용된 단어가 사용자의 질의 의도를 시스템에 명확하게 전달하지 못할 수 있기 때문이다.

예를 들어 “교통 사고가 났는데 어떻게 대처해야 하나요”와 같은 질의문은 피해자와 가해자중 누가 작성했느냐에 따라 서로 원하는 응답이 달라질 수 있다. 이처럼 사용자의 숨겨진 의도를 전달하는 것은 어렵기 때문에 사용자가 원하는 질의를 찾아내는 데에는 태생적인 한계가 존재한다. 따라서 주요 질의 응답 시스템은 사용자가 정보를 찾을 수 있도록 추가적으로 검색하는 보조 기능들을 제공한다.

검색 보조적인 기능으로서 Yahoo! Answers는 질의

응답이 포함된 카테고리의 다른 질의응답들을 보여준다[3]. Naver지식-IN과 Daum지식, 네이트 지식은 질의의 키워드와 일치하는 다른 질의응답들도 함께 보여준다. 그러나 이러한 기능들도 단순한 키워드 기반의 추가 검색이거나, 질의에 속한 카테고리의 질의들을 단순히 보여주는 정도로 그 기능이 매우 미약하다[6].

따라서 이러한 문제점을 개선하기 위해서, 검색된 질의응답이 다루는 정보를 내용적으로 포함하는 정도가 큰 질의응답들을 하위 질의응답으로서 추천하는 내용 기반 추천 방법이 제안되었다[1][16]. 내용적 포함관계를 계산하기 위해 퍼지 관계곱 연산자(Fuzzy Relational Product Operator)를 이용하여 사용자가 선택한 질의응답의 내용을 포함하는 정도가 높은 질의응답을 추천하였다. 내용적으로 포함하는 정도를 계산하기 위해 Kleene-Dienes 합의 연산자를 적용하여 퍼지 관계곱 연산자를 정의하였다. 그러나 Kleene-Dienes 합의 연산자는 정의 자체가 질의응답에 대한 설명의 의미적 포함관계를 고려한 방법이 아니기 때문에 질의응답의 의미적 포함 정도를 계산하기에 적합하지 않다.

이 문제를 해결하기 위해 본 논문에서는 새로운 합의 연산자인 의미적 퍼지함의 연산자(Semantic Fuzzy Implication Operator)를 정의하여 퍼지 관계곱 연산자에 적용함으로써 두 질의응답의 의미적 포함관계를 계산하는 방법을 제시한다.

이 방법의 장점은 다음과 같다. 사용자가 검색된 질의응답들의 내용에 대해 충분히 만족하지 못하는 경우에 그 중에서 사용자가 부분적으로 만족하는 질의응답을 선택하면, 선택된 질의응답에 대해서 의미적으로 포함하는 정도가 높은 하위 질의응답들을 추가로 찾아준다. 그러면 하위 질의응답 중에는 사용자가 원하는 내용이 있을 확률이 상대적으로 높아지게 된다. 사용자는 이러한 추천 과정을 반복하는 상호 작용을 통해 보다 적합한 질의응답을 찾아나가게 된다. 이 과정에서 새롭게 정의된 의미적 퍼지함의 연산자를 사용함으로써 이전의 퍼지함의 연산자에 비해서 의미적 포함관계가 보다 높은 질의응답을 찾아주어 사용자의 요구에 부합하는 결과를 찾아줄 확률을 보다 더욱 높일 수 있다는 장점이 있다.

II. 관련 연구

이전부터 사용자 질의에 대한 질의응답을 찾기 위한 많은 연구가 진행되고 있다. 초기의 질의 응답은 사실 질의(Factoid question)에 대한 응답을 열린 도메인(open-domain)에서 찾는 것이 목적이었다. 이후 질의에 대해 복잡한 응답을 요구하는 정의 질의(Definition question)에 대한 연구가 진행되었다[9][13][14]. 정의 질의는 사실 질의와 달리 한 개의 문장으로 응답할 수 없고, 응답 문장의 내용이 적합한지 쉽게 판단할 수 없는 특징을 갖는다.

최근에는 이용자들의 수가 상대적으로 매우 많은 웹 포털에 기반한 질의 응답 시스템이 등장하고 활발하게 사용되기 시작하면서 질의 응답 시스템에 대한 연구[2][3][6], 질의를 추천하는 방법에 대한 연구[4][5][16], 응답을 평가하는 방법에 대한 연구[10-12][15] 등 다양한 분야의 연구가 활발하게 이루어지고 있다.

질의 응답 시스템은 사용자의 질의에 대해 다른 사람 혹은 응답을 할 수 있을 정도의 전문가가 직접 응답을 입력하기 때문에 일반적으로 좋은 응답을 제공할 수 있다. 게다가 질의 응답 시스템은 단순히 질의와 응답의 내용을 저장하는 기능을 제공할 뿐 아니라 사용자의 활동 정보를 다루는 프로필, 응답이 질의에 적절했는가를 평가하는 기능 등 보다 다양한 정보를 이용하여 응답의 질을 향상시킬 수 있도록 노력하고 있다.

질의 응답 시스템에 저장되는 질의응답의 수와 내용이 많아짐에 따라 사용자의 의도에 적합한 질의를 추천하는 방법이 연구되었다[4][5][16]. Hu는 사용자 모델링 성분을 이용한 균형 질의 추천 방법을 제안하였다. 사용자 모델링 방법은 각 사용자들의 흥미분야와 전문 영역을 평가하는데 사용하였다[4]. Liu는 단어 연관 모델(word relevance model)을 이용하여 질의 응답 시스템에서 유사한 질의를 찾는 방법을 제안하였다[5]. 단어 연관 모델은 특정 단어와 의미적으로 연관 정도가 높은 단어를 찾는 데 활용된다.

이러한 연구들은 사용자의 질의에 대한 최선의 응답을 찾아주기 위한 연구이다. 하지만 항상 사용자를 만족시킬 수 있는 응답을 제공할 수 있는 것은 아니다. 그

래서 사용자가 응답 내용에 만족하지 못할 경우 사용자로 하여금 만족스러운 응답을 찾을 수 있도록 보조하는 기능들에 대한 연구가 이루어졌다.

Yahoo! Answer는 사용자가 질의를 입력하는 단계에서 질의에 적합한 카테고리를 선택하도록 한다. 이후 이 질의가 검색되면 질의의 키워드를 포함하는 다른 질의들(Related Questions)과 질의가 속한 카테고리의 상위 카테고리라 동위 카테고리를 열람할 수 있도록 카테고리들(Categories)을 보여준다. 그리고 카테고리에서 아직 응답이 없는 질의들(Open Questions)과 응답이 완료된 다른 질의응답들(Resolved Questions)을 제공한다[3].

네이버 지식-IN, 다음 지식, 네이버 지식은 대표적인 한국어 질의 응답 시스템이다. 이들은 입력된 질의의 키워드가 포함된 다른 지식들을 보여준다[6]. 예를 들어 네이버 지식-IN은 질의 보조적인 방법으로 질의의 검색 결과를 정확도순 또는 등록순으로 보여줌으로써 제공한다. 그리고 실시간으로 등록되는 질의들을 보여줌으로써 다른 사용자로 하여금 빠르게 응답을 할 수 있는 환경도 제공한다. 그러나 이 시스템들은 키워드 기반의 검색을 사용하기 때문에 같은 키워드를 갖는다고 하더라도 의도가 다른 질의응답들이 검색되거나, 의미와 상관없이 단순히 질의에 속한 카테고리의 질의들을 보여주는 정도로 그 기능이 매우 미약하다는 단점이 있다.

III. 질의 추천 과정

이 절에서는 제안하는 질의 추천 과정의 흐름을 요약한다. 질의 응답 시스템은 질의 응답들을 질의-응답 데이터베이스(이후 *KDB*로 표기)에 저장하며 하나의 지식 k 는 하나의 질의 Q 와 질의에 대한 응답 A 들로 이루어져 있다. ($k = \{Q, A_1, A_2, \dots, A_n\}$)

본 논문에서 구현한 질의 응답 시스템이 추천 지식을 검색하는 구체적인 프로세스는 다음과 같다.

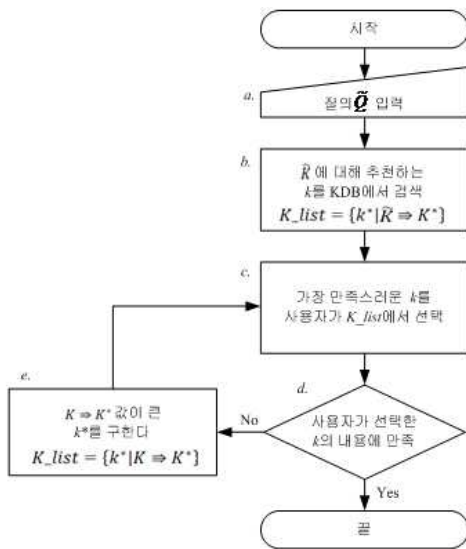


그림 1. 지식 추천 과정

사용자 질의를 Q 라 하고, KDB에 저장된 지식을 k 라 하자. KDB는 지식 k들을 갖는 집합 $KDB = \{k_1, k_2, \dots, k_i, \dots, k_m\}$ 이다. 사용자는 [그림 1] 과 같은 과정을 수행하면서 추천 지식을 얻을 수 있다.

[그림 1]의 a에서 사용자가 질의 Q를 입력하면 [그림 1]의 b에서 시스템은 Q의 퍼지 집합 R의 추천 질의가 될 수 있는 k를 KDB에서 검색하여 K_list에 저장한다. [그림 1]의 c에서 사용자는 K_list의 k들 중에서 질의에 가장 만족스러운 정보를 가졌을 것으로 생각하는 임의의 k를 주관적으로 판단하여 선택한다. [그림 1]의 d에서 사용자가 k에 포함된 응답 내용에 만족할 경우 질의 추천 과정은 종료된다. 그렇지 않을 경우, [그림 1]의 e에서 사용자가 선택한 k와 의미적 포함 관계 값이 높은 k^* 를 K_list에 저장한다. 의미적 포함 관계값 $K \Rightarrow K^*$ 이 높은 k^* 는 선택한 k가 다루는 내용을 더 잘 설명하고 자세히 설명하는 것을 의미한다. 시스템은 $K \Rightarrow K^*$ 값이 큰 순서대로 지정한 개수만큼의 k^* 를 K_list에 저장한다. 의미적 퍼지 관계 곱 연산자 \Rightarrow 는 IV.2절에서 설명한다.

의미적 퍼지 관계곱 연산자 $K \Rightarrow K^*$ 는 두 지식 사이의 의미적 포함 관계를 계산한다. 사용자는 더 이상 다른 질의를 검색하지 않아도 된다고 생각할 때 까지 [그

림 1]의 c부터 [그림 1]의 e까지의 과정을 반복한다. 이 판단은 같은 질의라도 입력한 사용자에 따라 다르다. 추천 지식을 찾아가는 과정이 반복될 때 마다 이전 단계의 지식을 의미적으로 포함하는 지식들이 K_list에 저장된다. 그렇기 때문에 사용자가 원하는 내용을 찾을 수 있을 가능성이 점점 높아진다.

예를 들어 제안하는 방법을 이용할 경우, 검색된 지식들 중에서 부분적으로 만족스러운 내용이 있을 것으로 예상되는 지식들을 보여준다. 사용자는 그 중에서 "What kinds of methods for windows programming?"을 선택한다고 가정하고 이 질의에 대한 응답에는 "JAVA uses AWT or SWING for windows programming"라는 내용이 있다고 하자. 제안하는 방법은 이 지식의 내용을 의미적으로 포함하는 정도가 높은 값을 갖는 다른 질의 "JAVA AWT programming method" 나 "What is the JAVA SWING?" 들에 대한 지식들을 넓은 의미를 갖는 지식으로서 자동으로 추천한다. 따라서 사용자는 새로운 질의를 다시 입력하지 않고도 추천과정을 반복하면 자신이 원하는 정보를 찾아갈 수 있다.

IV. 퍼지 함의 관계 곱

1. 퍼지 관계 곱

퍼지 관계 곱(Fuzzy Relational Product)은 여러 단계의 퍼지 관계 연산으로 정의된다. 예를 들면 "A는 B이고 B가 C이면, A는 C이다"라는 삼각 논리를 퍼지 관계 논리로 바꾸면 "A가 B에 관련되고 B가 C에 관련되면 A는 C에 관련된다"로 정의할 수 있다. 단 퍼지값은 크리스프값과 다르기 때문에 퍼지 관계 곱은 다음과 같이 정의된다.

(정의 1) 유한집합 $U1, U2, U3$ 이 있다고 가정하고 $U1$ 에서 $U2$ 로의 퍼지 관계를 R , $U2$ 에서 $U3$ 로의 퍼지 관계를 S 라 하자. 즉, R 은 $U1 \times U2$ 의 부분집합이고 S 는 $U2 \times U3$ 의 부분집합이다. 퍼지 관계 곱은 $a \in U1$ 이고 $c \in U3$ 일 때 a 가 c 에 관련되는 정도를 나타낸다. 이때 a

에서 c 로의 퍼지 관계를 표현하는 삼각 논리 곱 \triangle 은 다음과 같이 정의된다.

$$(RAS)_{ik} = \frac{1}{N_j} \sum_j (R_{ij} \rightarrow S_{jk}) \quad (1)$$

\rightarrow 은 퍼지 함의 연산자를 의미하며 대표적으로 식 2와 같은 Kleene-Dienes 퍼지 함의 연산자가 있다[7].

$$a \rightarrow b = (1 - a) \vee b = \max(1 - a, b) \quad (a = 0 \sim 1, b = 0 \sim 1) \quad (2)$$

2. 의미적 퍼지 관계 곱

추천하는 지식은 어떤 정보에 대해 사용자가 찾은 지식보다 더 잘 설명하고 자세히 설명할 수 있어야 한다. 지식을 퍼지 집합으로 간주하고 지식의 내용을 퍼지 집합의 멤버라고 하면 그 내용이 가진 정보가 지식을 잘 설명하는 정도를 퍼지 집합의 멤버십 정도(membership degree)로 나타낼 수 있다. 높은 멤버십 정도 값은 지식의 내용이 가진 정보가 지식을 잘 설명하고 있음을 의미한다. 어떤 멤버가 지식을 잘 설명하는 정도를 정의하고, 모든 멤버가 처음 지식과 추천 지식에 대해 설명하는 정도를 얻으면, 두 지식 간의 의미적 포함 정도를 퍼지 함의 연산으로 계산할 수 있다.

$KDB = \{k_1, k_2, \dots, k_i, \dots, k_m\}$ 의 i 번째 지식 k_i 는 $k_i = \{Q_i, A_1^i, A_2^i, \dots, A_{n_i}^i\}$ 로 정의한다. Q_i 는 k_i 의 질문에 해당되는 내용이고, A_l^i 는 Q_i 에 대한 l 번째 응답 내용이다. K_i 는 k_i 에 대한 퍼지 집합이고 K_i 의 멤버 T 는 임의의 질의-응답 쌍이며 T 의 멤버십 정도는 T 가 k_i 를 잘 설명하는 정도이다. 이러한 퍼지 집합 K_i 는 식 3과 같이 정의한다.

$$K_i = \left\{ (T_{ip}, \mu_{K_i}(T_{ip})) \mid \begin{array}{l} T_{ip} = \{(Q_i, A_l^i)\}, \\ k_i = \{Q_i, A_1^i, A_2^i, \dots, A_{n_i}^i\} \in KDB, \\ l = 1, 2, \dots, m, p = 1, \dots, n_i \end{array} \right\} \quad (3)$$

여기서 T_{ip} 는 지식 k_i 에 속한 p 번째 질의-응답 쌍이다. KDB 에 저장된 모든 지식들에 대한 질의-응답 쌍의 집합 TDB 는

$$TDB = \{T_{11}, T_{12}, \dots, T_{1n_1}, T_{21}, T_{22}, \dots, T_{2n_2}, \dots, T_{m1}, T_{m2}, \dots, T_{mn_m}\} \quad (4)$$

으로 나타낼 수 있다.

퍼지 집합을 구성하는 멤버십 정도는 퍼지 집합의 지식과 각 T_{ip} 사이의 확장된 유사도($esim$)[1]를 이용하

며 다음과 같이 정의한다.

$$\mu_{K_i}(T_{ip}) = esim(Q_i, T_{ip}) \quad (5)$$

여기서 $esim(Q_i, T_{ip})$ 는 유사도 계산을 의미하며 높은 유사도 값은 T_{ip} 가 Q_i 를 잘 설명하는 정도를 의미한다[1][8].

어떤 지식 k_i 가 다른 지식 k_j 에 의미적으로 포함되는 정도를 계산하기 위해 의미적 퍼지 관계 곱 연산자(\Rightarrow , Semantic Fuzzy Relational Product Operator)를 식 6과 같이 정의한다.

$$K_i \Rightarrow K_j = \frac{1}{|TDB|} \sum_{T^r \in TDB} (\mu_{K_i}(T^r) \rightarrow \mu_{K_j}(T^r)) \quad (6)$$

여기서 연산자 \rightarrow 는 의미적 퍼지 함의 연산자(Semantic Fuzzy Implication Operator)이고 V.3절에서 정의한다. T^r 은 TDB 의 원소들을 일렬로 나열했을 때(serialize) r 번째 T 를 의미하며

$$TDB = \{T_{11}, T_{12}, \dots, T_{1n_1}, T_{21}, T_{22}, \dots, T_{2n_2}, \dots, T_{m1}, T_{m2}, \dots, T_{mn_m}\} \\ = \{T^1, T^2, \dots, T^s\}, \quad s = \sum_{v=1}^m n_v$$

이다.

$K_i \Rightarrow K_j$ 는 TDB 의 모든 T^r 에 대한 $\mu_{K_i}(T^r) \rightarrow \mu_{K_j}(T^r)$ 값들의 평균으로 계산되는 의미적 퍼지 관계 곱 연산이다. 즉 $K_i \Rightarrow K_j$ 값은 지식 k_i 가 k_j 에 의미적으로 포함되는 정도를 의미한다.

3. 의미적 퍼지 함의 연산자

3.1 대소 관계 규칙

사용자에게 유용한 추천 질의의 조건은 첫째, 추천되는 질의는 보다 많은 정보를 찾을 수 있어야 한다. 추천 질의에 대한 질의응답의 정보량은 사용자가 입력한 질의에 대한 질의응답의 정보량보다 더 많아야 한다. 둘째, 추천 질의는 찾고자 하는 정보와 연관 정도가 높은 정보를 찾을 수 있어야 한다. 추천 질의에 대한 질의응답이 정보량이 많아야 할 뿐 아니라 사용자가 입력한 질의에 대한 응답이 될 수 있어야 한다.

이전 연구에서 사용한 퍼지 함의 연산자는 \rightarrow 는 Kleene-Dienes 퍼지 함의 연산자로 $a \rightarrow b = \max(1 - a, b)$ 와 같다. 그런데 Kleene-Dienes 함의 연산자는 지식의 의미적 포함 관계를 계산하기 위해 제안된 방법이 아니기 때문에 의미적으로 포함하는 정도를

계산하는데 적합하지 않다.

지식 설명의 멤버십 정도 값 $a = \mu_K(T), b = \mu_{K'}(T)$ 는 각각 지식 K 와 K' 에 대해 T 가 설명하는 정도를 나타낸다. $a \rightarrow b$ 의 값은 T 에 의해서 K' 이 K 를 얼마나 의미적으로 포함하느냐를 나타내는 정도를 계산하는 것이다. 그런데 a 와 b 의 값이 작은 값 일때 $a \rightarrow b$ 가 의미하는 것은 T 가 K 에서 중요하지 않은 내용인데 K' 에서도 T 가 중요하지 않게 간략히 서술되었다는 뜻이다. 이 경우에 K' 이 K 를 의미적으로 포함하는 정도를 T 로써 판단할 수 있는 근거가 매우 약하다. 따라서 이 경우 $a \rightarrow b$ 값이 작은 값이어야 한다. 그런데 Kleene-Diense는 큰 값으로 계산한다. 이런 의미에서 Kleene-Diense 연산자는 의미적 포함관계를 계산하는데 있어서 적합하지 않다. 따라서 새로운 의미적 함의 연산자 \rightarrow 를 정의한다.

$x_a = \mu_{K_a}(T^r)$ 이고 $y_b = \mu_{K'_b}(T^r)$ 라고 하자. 의미적 퍼지 함의 연산자의 값 $x_a \rightarrow y_b$ 의 의미는 K_a 가 K'_b 에 설명된 T^r 로써 의미적으로 포함되는 정도를 나타낸다. 이때 K_a 와 K'_b 는 각각 T^r 로써 설명되는 정도이다. 즉 T^r 이 K_a 를 설명하는 정도보다 T^r 이 K'_b 를 더 잘 설명되거나 더 많이 설명된다는 정도를 계산한다.

$x_a \rightarrow y_b$ 값이 가질 수 있는 상황은 [그림 2]와 같고 그에 따른 의미는 [표 1]과 같다.

y_1	v_2	v_1
y_0	v_4	v_3
y_b x_a	x_0	x_1

$$v_r = x_a \rightarrow y_b$$

$$x_0 = \mu_{K_0}(T), x_1 = \mu_{K'_1}(T)$$

$$y_0 = \mu_{K'_0}(T), y_1 = \mu_{K'_1}(T)$$

$$|x_1 - x_0| = |y_1 - y_0|, (x_0 < x_1, y_0 < y_1)$$

그림 2. $x_a \rightarrow y_b$ 값의 종류

표 1. 의미적 퍼지 함의 연산자의 의미

x_a	y_b	$v_r = x_a \rightarrow y_b$	$x_a \rightarrow y_b$ 의 의미
x_1 (큰값)	y_1 (큰값)	$v_1 = x_1 \rightarrow y_1$	T 가 K_1 를 설명하는 중요한 내용이며 K'_1 에서도 T 가 중요하고 자세히 서술된 내용이다.
x_0 (작은값)	y_1 (큰값)	$v_2 = x_0 \rightarrow y_1$	T 가 K_0 에서 중요하지 않은 내용인데도 불구하고 K'_1 에서 T 가 중요하고 자세히 서술된 내용이다.
x_1 (큰값)	y_0 (작은값)	$v_3 = x_1 \rightarrow y_0$	T 가 K_1 에서 중요한 내용인데도 불구하고 K'_0 에서는 T 가 중요하지 않게 간략하게 서술되었다.
x_0 (작은값)	y_0 (작은값)	$v_4 = x_0 \rightarrow y_0$	T 가 K_0 에서 중요하지 않은 내용이기에 때문에 K'_0 에서도 T 가 중요하지 않게 간략하게 서술되었다.

x_0 는 x_1 에 비해서 작은 값이다. 값이 작다는 의미는 T 가 지식 k_0 를 설명하는데 있어서 별로 중요하지 않은 내용이거나 k_0 를 자세하게 서술한 내용이 아니거나 k_0 를 지나치게 간략하게 서술한 내용임을 의미한다. 다른 관점에서 보면 x_1 은 x_0 에 비해서 큰 값이다. 값이 크다는 의미는 T 가 지식 k_1 를 설명하는데 있어서 중요한 내용이거나 k_1 에 대해 자세히 서술되거나 충분히 서술되었음을 의미한다. y_0 와 y_1 의 경우는 x_0 와 x_1 의 경우와 마찬가지로이다.

k_i 가 k_j 에 의미적으로 포함된다는 것은 k_i 에서 설명되는 내용은 k_j 에서도 설명되고 있음을 의미한다. 의미적으로 포함되는 정도가 높을수록 제공하는 정보의 범위가 넓으므로 사용자의 질의의도에 대한 답변일 가능성이 높아지므로 좋은 추천지식이 될 수 있다.

의미적 퍼지 함의 연산자의 값의 크기를 정의하기 위해서는 먼저 [그림 4]에 있는 v_1, v_2, v_3, v_4 의 상대적 크기를 비교하여야 한다. 이를 위해서 먼저 v_1 과 v_2 의 크기를 비교하고 v_3 과 v_4, v_2 과 v_3 들에 대해서도 각각 크기를 비교하여 전체적인 크기의 순서를 다음과 같이 정한다.

v_1 과 v_2 의 크기를 비교하자, T 가 k_1 를 설명하는데 있어서 중요한 내용인데 반해서, k_0 을 설명하는데 있어서 덜 중요한 내용이다. T 는 k'_1 에서 중요한 내용이다. 즉 k_1 에서 중요한 내용이 k'_1 에서도 중요하게 서술되고 있고 k_0 에서 덜 중요한 내용이 k'_1 에서는 중요하게 서술되고 있는 것이다. 따라서 중요한 내용이 자세하고 중요하게 설명되는 경우의 의미적 포함 정도가 덜 중요한 내용을 자세하고 중요하게 설명하는 경우의 의미적 포함 정도 보다 더 크다고 말할 수 있다. 따라서 v_1 이 v_2 보다 더 큰 값을 가져야 한다고 볼 수 있다.

v_3 와 v_4 의 크기를 비교하자. 공히 T 는 k'_0 에 대해서 간략하게 설명하고 있다. 그런데 T 는 k_1 에서는 중요한 내용이고 k_0 에서는 덜 중요한 내용이다. 즉 k_1 에서 중요한 내용이지만 k_0 에서는 덜 중요한 내용을 k'_0 에서

같은 강도로 설명하고 있다. 따라서 k_1 에서 중요한 내용을 k'_0 에서 의미적으로 포함하는 정도는 k_0 에서 덜 중요한 내용을 k'_0 에서 의미적으로 포함하는 정도보다 더 크다고 할 수 있다. 따라서 v_3 가 v_4 보다 더 큰 값을 가져야 한다고 볼 수 있다.

v_2 와 v_3 의 크기를 비교하자. T 는 k_0 에서 덜 중요한 내용인데 k'_1 에서는 중요한 내용으로서 서술되고 있다. 반대로 T 는 k_1 에서는 중요한 내용인데도 k'_0 에서는 중요하지 않게 서술되고 있다. 즉 처음 지식에서 중요하지 않은 내용을 다른 지식에서 중요하게 서술되면 다른 지식은 더 많은 내용을 추가적으로 포함한다고 말할 수 있다. 그러나 처음 지식에 있어서 중요한 내용인데도 불구하고 다른 지식에서 소홀하게 서술하면 다른 지식은 상대적으로 중요한 내용을 누락시킬 수 있다. 따라서 덜 중요한 내용을 중요하게 자세히 서술하는 경우의 의미적 포함 정도가 중요한 내용을 소홀히 서술하는 경우의 의미적 포함 정도 보다 더 크다고 간주할 수 있다. 따라서 v_2 가 v_3 보다 더 크다고 할 수 있다.

최종적으로 $v_1 > v_2, v_3 > v_4$ 이고 $v_2 > v_3$ 이고 $v_3 > v_4$ 이므로 네 값의 대소 관계는 $v_1 > v_2 > v_3 > v_4$ 와 같다고 결론지을 수 있다. 이를 의미적 퍼지 함의 연산자 대소 관계 규칙이라고 정의한다.

3.2 의미적 퍼지 함의 값 행렬

본 논문에서는 위의 의미적 퍼지 함의 연산자 대소관계 규칙을 바탕으로 의미적 퍼지 함의 연산자 값 행렬 H_n 을 생성한다. 여기서 n 은 구간의 개수를 의미한다. 예를 들어 [그림 3]의 b 의 n 은 2가 된다.

2	4	2	1	1	0.625	0.875	1.0
1	7	5	3	0.5	0.25	0.5	0.75
0	9	8	6	0	0	0.125	0.375
b / a	0	1	2	y_b / x_a	0	0.5	1

a) 크기 순서 행렬 M_2 b) 의미적 퍼지 함의 값 행렬 H_2

그림 3. 의미적 퍼지 함의 값 행렬의 구성

[그림 3]의 a 는 V.3.2절에서 정의한 의미적 대소 관계 규칙을 만족하는 크기 순서 행렬이다. 의미적 퍼지 함의 연산자 행렬은 [그림 3]의 b 와 같이 행과 열 값에 대한 행렬값이 의미적 퍼지 함의 연산자 대소관계 규칙을 만족하면서, 행렬값과 행과 열을 일정한 차이를 갖는 0과 1 사이의 값으로 구성된 행렬이다. 완성된 행렬 H 의 행 y_b 와 열 x_a 에 대한 행렬값은 $x_a \mapsto y_b$ 로 나타낸다. 이 행렬 값이 의미적 퍼지 함의 연산자 값이다.

입력의 입력 값 x, y 에 대한 $x \mapsto y$ 값을 구하기 위해 x 와 y 를 크기 순서 행렬 $M_n(a, b)$ 의 a, b 로 근사시킨다. a, b 는 각각 0부터 n 까지의 자연수로 나타낸다.

식 7은 [그림 3]의 a 와 같이 a, b 에 대한 행렬값을 구하는 식이다.

$$M_n(a, b) = \begin{cases} ((n+1)^2 + 1) - M_n^*(a, b) & \text{if } (a+b) \leq n \\ M_n^*(n-a, n-b) & \text{if } (a+b) > n \end{cases}$$

where $M_n^*(a, b) = \frac{(a+b)(a+b+1)}{2} + (b+1)$ (7)

식 8은 [그림 3]의 a 의 M 행렬값으로 [그림 3]의 b 의 행렬값을 구하는 식이다.

$$G_n(a, b) = \frac{(n+1)^2 - M_n(a, b)}{(n+1)^2 - 1} \quad (8)$$

퍼지 함의 연산자 행렬의 행렬값을 얻는 이산적인 함의연산자 함수인 $H_n(i, j)$ 는 여러가지로 정의할 수 있다. 예를 들어 $H_n(i, j) = G_n(i, j), H_n(i, j) = G_n(i, j)^2,$

$$H_n(i, j) = \tan\left(\frac{\pi}{4} \cdot G_n(i, j)\right), \quad H_n(i, j) = \sqrt{G_n(i, j)},$$

$$H_n(i, j) = \sin\left(\frac{\pi}{2} \cdot G_n(i, j)\right) \text{ 등으로 정의할 수 있다. 우}$$

리는 단순히 $H_n(x_i, y_j) = G_n(i, j)$ 를 이용한다. 따라서 $x \mapsto y$ 에 대한 의미적 퍼지 함의 값 행렬(Semantic fuzzy implication value matrix) 값을 계산하는 방법은 다음과 같이 요약된다.

$$x \mapsto y \approx H_n(x_a, y_b) = H_n\left(\frac{1}{n} \cdot \left\lceil \frac{2nx-1}{2} \right\rceil, \frac{1}{n} \cdot \left\lceil \frac{2ny-1}{2} \right\rceil\right) = G_n\left(\left\lceil \frac{2nx-1}{2} \right\rceil, \left\lceil \frac{2ny-1}{2} \right\rceil\right) \quad (9)$$

1	0.54	0.63	0.7	0.77	0.83	0.88	0.92	0.95	0.98	0.99	1
0.9	0.45	0.53	0.62	0.69	0.76	0.82	0.87	0.91	0.94	0.97	0.98
0.8	0.37	0.44	0.53	0.61	0.68	0.75	0.81	0.86	0.9	0.93	0.96
0.7	0.29	0.36	0.43	0.52	0.6	0.68	0.74	0.8	0.85	0.89	0.93
0.6	0.23	0.28	0.35	0.43	0.51	0.59	0.67	0.73	0.79	0.84	0.88
0.5	0.17	0.22	0.28	0.34	0.42	0.5	0.58	0.66	0.73	0.78	0.83
0.4	0.12	0.16	0.21	0.27	0.33	0.41	0.49	0.58	0.65	0.72	0.78
0.3	0.08	0.11	0.15	0.2	0.26	0.33	0.4	0.48	0.57	0.64	0.71
0.2	0.04	0.07	0.1	0.14	0.19	0.25	0.32	0.39	0.48	0.56	0.63
0.1	0.02	0.03	0.06	0.09	0.13	0.18	0.24	0.31	0.38	0.47	0.55
0	0	0.01	0.03	0.05	0.08	0.13	0.18	0.23	0.3	0.38	0.46
$\frac{1}{n}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

그림 4. 의미적 퍼지 함의 값 행렬의 예(H_{10})

예를 들어 $n=10$ 인 의미적 퍼지 함의 연산자 행렬은 H_{10} 을 구축할 경우 [그림 4]와 같다. 이 행렬을 이용하면 $0.3113 \mapsto 0.864$ 에 대한 값은 $H_{10}(0.3113, 0.864) = G_{10}(3, 9) = 0.69$ 가 된다.

V. 실험 및 평가

본 논문에서 사용된 실험 데이터로서의 지식은 Yahoo! Answers의 데이터를 WebCrawler 프로그램을 사용하여 자동으로 수집한 후 KDB에 저장하였다. 수집된 지식은 하나의 질의문장과 그에 대한 여러 개의 응답들로 구성된다. KDB는 이러한 방식으로 랜덤하게 수집된 16000여 개의 지식들로 구성하였다.

같은 질의를 입력하더라도 질의에 숨겨진 의도는 사용자에 따라 그 의도가 달라질 수 있다. 그렇기 때문에 추천 질의에 대한 평가를 하기 위해, 추천 질의가 사용자의 질의에 연관되는지 여부를 전문가나 실험참여자가 직접 판정해야 한다[5][9].

본 실험에서는 추천된 질의가 원래 질의에 연관되었다고 여겨지는 개수를 평가 기준으로 하였다. 연관되는지 여부는 질의를 입력한 사용자가 직접 판단한다. 실험을 위해 각 방법에서 추천되는 질의의 개수를 동일하게 한 후 그 중에서 사용자를 만족시키는 추천 질의의 비율을 구한다. 그리고 실험의 검증을 위해 t-검정을 사용한 가설 검정으로 각 방법들에 대한 비교 우위를 검증하였다. 첫 번째 실험은 제안하는 방법을 사용하여 추천 과정을 반복할수록 추천된 질의의 성능이 좋아짐

을 보인다.

표 2. 추천 과정에 대한 질의의 성능 비교(단위:%)

	FRP-KD-1	FRP-SFIO-1	FRP-KD-2	FRP-SFIO-2
질의 1	24.5	25.0	30.5	39.0
질의 2	11.5	11.5	24.5	35.5
질의 3	15.5	36.0	15.0	37.5
질의 4	6.5	10.5	10.0	27.5
질의 5	24.5	16.0	15.0	26.0
질의 6	24.5	40.0	25.0	41.0
질의 7	20.5	21.5	18.5	18.0
질의 8	12.0	22.0	31.0	26.0
질의 9	17.0	13.0	15.5	14.5
질의 10	18.5	11.0	16.0	11.5
질의 11	23.5	23.5	21.5	21.0
질의 12	25.0	25.0	25.5	24.5
질의 13	19.5	19.0	25.0	39.0
질의 14	19.5	47.0	18.0	43.5
질의 15	11.5	14.0	12.5	41.5
질의 16	13.0	34.5	13.0	27.0
질의 17	7.0	19.0	18.0	21.0
질의 18	18.0	14.0	19.0	49.0

[표 2]는 Kleene-Dienes 연산자를 이용하였을 때 [1][16]와 SFIO(의미적 퍼지 함의 연산자) 연산자를 이용하였을 때, 처음 질의에 연관된 것으로 판단된 추천 질의의 비율을 비교한 결과이다. 세 번 이상의 추천 과정을 거칠 경우 이전 단계에서 추천되었던 질의가 다시 추천되는 경우도 많기 때문에 두 번의 추천 과정만 비교하였다.

두 번째 실험은 제안하는 질의 추천 방법의 우수성을 보이기 위해 질의 응답 시스템에 사용되는 다른 방법들과 비교하는 실험을 진행하였다.

표 3. 각 추천 방법으로 얻은 질의의 성능 비교(단위:%)

	VSM	Kernel	FRP-SFIO
질의 1	21.0	19.0	24.0
질의 2	23.0	36.5	28.0
질의 3	12.5	20.5	25.0
질의 4	20.5	7.5	15.0
질의 5	10.5	10.5	14.0
질의 6	21.0	14.5	24.0
질의 7	15.0	14.5	16.0
질의 8	14.0	20.0	27.0
질의 9	15.0	13.5	16.0
질의 10	15.5	22.0	30.5
질의 11	22.5	22.0	29.0
질의 12	19.0	31.5	38.5
질의 13	26.5	26.5	32.0
질의 14	16.0	18.5	25.5
질의 15	18.0	17.5	23.0

[표 3]은 벡터 공간에서 코사인 유사도를 이용한 질의 추천 방법(VSM), Liu가 제안한 Kernel 방법[5], 의미적 퍼지 함의 연산자를 이용한 추천방법을 비교한 결과이다.

두 실험의 검증은 위해 다음과 같이 비교 기준을 설정하였다.

비교 1: 첫 번째 실험에서 FRP-KD-1 방법과 FRP-SFIO-1 방법을 비교.

비교 2: 첫 번째 실험에서 FRP-SFIO-1 방법과 FRP-SFIO-2 방법을 비교.

비교 3: 두 번째 실험에서 VSM 방법과 FRP-SFIO-1 방법을 비교.

비교 4: 두 번째 실험에서 Kernel 방법과 FRP-SFIO-1 방법을 비교.

예를 들어 비교 1을 검증하기 위한 순서는 다음과 같다. 귀무가설 $H_0: \mu_1 \leq \mu_2$ 을 “FRP-SFIO-1방법이 FRP-KD-1방법보다 성능이 우수하지 않다.”고 정하고 대립가설 $H_1: \mu_1 > \mu_2$ 을 “FRP-SFIO-1방법이 FRP-KD-1방법보다 성능이 우수하다.”고 정한다. t -검정에 필요한 값들은 [표 2]에서 다음과 같이 계산된다. FRP-SFIO-1방법과 FRP-KD-1방법의 평균은 각각 $X_1 = 22.5$, $X_2 = 17.5$ 이고 표준편차는 각각 $S_1 = 10.7$, $S_2 = 6.0$ 이다. 표본의 크기는 $n_1 = n_2 = 18$ 으로 같으므로 검정통계량 t 와 자유도 f 는 다음과 같이 계산된다.

$$\text{검정 통계량 } t = \frac{(X_1 - X_2)}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} = 1.730 \quad (10)$$

$$\text{자유도 } f = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{\frac{(S_1^2/n_1)^2}{(n_1 - 1)} + \frac{(S_2^2/n_2)^2}{(n_2 - 1)}} = 26.8 \quad (11)$$

H_0 를 기각하고 H_1 를 수락하려면 $t > t_\alpha(f)$ 이어야 한다. $t > t_{0.05}(27) = 1.703$ 이므로 H_0 를 기각하고 H_1 를 수락한다. 즉 FRP-SFIO방법이 Kernel방법에 비해 유의수준 5% 이내로 더 우수하다고 결론내릴 수 있다.

이 검증 과정을 각 비교 방법에 적용한 내용은 다음 [표 4]와 같다.

표 4. 각 비교 기준에 대한 t-검정 계산 결과

	비교 1	비교 2	비교 3	비교 4
모평균 μ_1	FRP-SFIO-1	FRP-SFIO-2	FRP-SFIO	FRP-SFIO
모평균 μ_2	FRP-KD-1	FRP-SFIO-1	VSM	Kernel
귀무가설 H_0	$\mu_1 \leq \mu_2$	$\mu_1 \leq \mu_2$	$\mu_1 \leq \mu_2$	$\mu_1 \leq \mu_2$
대립가설 H_1	$\mu_1 > \mu_2$	$\mu_1 > \mu_2$	$\mu_1 > \mu_2$	$\mu_1 > \mu_2$
평균 X_1	22.5	30.4	24.7	24.7
평균 X_2	17.5	22.5	18.2	19.8
표준편차 S_1	10.7	10.9	7.0	7.0
표준편차 S_2	6.0	10.7	4.4	7.7
표본 크기 n_1	18	18	15	15
표본 크기 n_2	18	18	15	15
검정 통계량 t	1.730	2.193	3.04	1.81
자유도 f	26.8	34.0	23.69	27.8
$t_{0.05}(f)$	1.703	1.691	1.711	1.701
$t > t_\alpha(f)$ 결론	H_0 기각	H_0 기각	H_0 기각	H_0 기각

VSM 방법은 두 질의 사이에 같은 단어가 사용되지 않는 경우 그 결과 값이 0이 된다. 그런데 질의 응답 시스템의 데이터들은 질의 문장이 짧은 경우가 많이 있다. VSM방법은 추천 질의를 찾기 위해 문장 사이의 유사도만을 계산하기 때문에 많은 응답을 찾아내지 못한다. Kernel 방법은 질의 응답 시스템에 저장된 모든 질의들을 이용하여 사용자가 입력한 질의와 같이 나타난 단어들을 찾아낸다. 그리고 찾아낸 단어들을 이용하여 질의를 확장한다. 그런 면에서 Kernel 방법은 VSM방법보다 더 우수한 면이 있다. 그러나 두 방법 모두 키워드 기반의 검색 방법으로서 주어진 단어와 같은 단어를 많이 포함하는 질의를 찾아주는 방법이다.

그러나 본 논문에서 제안한 FRP-SFIO방법은 단순히 같은 키워드를 포함하는 질의 및 응답을 찾는 것이 아니라, WordNet을 활용하여 질의에 포함된 검색 단어를 확장하고, 확장 유사도 계산 방법을 적용하여 짧은 문장일지라도 유사도 계산을 가능하게 해주며, 문장들의 의미적 포함 관계를 퍼지기법으로 계산하여 적용하기 때문에, 주어진 질의문에 대한 기존의 내용보다 좀 더 폭 넓은 의미를 가지는 질의 응답 들을 찾아 줄 수 있다. 따라서 사용자가 질의에서 표현하지 않았던 숨겨진 의도가 있었다면 추천된 질의 중에서 자신의 의도와 일치하는 것을 발견할 확률을 더 높여준다.

VI. 결론

본 논문에서는 추천되는 지식의 성능을 개선하기 위해 의미적 퍼지 관계곱 연산으로 지식을 추천할 때 이용하는 의미적 퍼지 함의 연산자를 정의하였다. 본 논문에서 제안하는 방법은 사용자가 원하는 응답을 얻기 위한 절차를 줄여주고 사용자의 의도에 부합하는 응답을 찾는 과정을 돕는다.

실험을 통해 추천 과정을 반복할수록 질의의 성능이 좋아짐을 보였다. 그리고 기존 질의 추천방법과 비교를 통해 경쟁력있는 결과를 보였다. 향후 연구에서는 실험 데이터의 전문화와 평가 요소의 다변화를 통해 실험 평가의 주관적 요소를 줄이는 방향으로 개선시킬 예정이다.

참 고 문 헌

- [1] C. M. Ahn, J. H. Lee, B. G. Choi, and S. Park, "Question Answering System with Recommendation using Fuzzy Relational Product Operator," Proc. of Information Integration and Web-based Applications & Services, pp.853-856, 2010.
- [2] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the right facts in the crowd: factoid question answering over social media," Proc. of the 17th international conference on World Wide Web, 2008.
- [3] Z. Gyöngyi, G. Koutrika, J. Pedersen, and H. Garcia-Molina, "Questioning Yahoo! Answers," First Workshop on Question Answering on the Web at the 17th International World Wide Web Conference, 2008
- [4] D. Hu, S. Wang, L. Wenyin, and E. Chen, "Question recommendation for user-interactive question answering systems," Proc. of the 2nd int. conf. on Ubiquitous information management and communication, pp.39-44, 2008.
- [5] M. Liu, Y. Liu, and Q. Yang, "Searching semantically similar questions from a large community based question archive," International Conference on Natural Language Processing and Knowledge Engineering, pp.1-8, 2009.
- [6] K. K. Nam, M. S. Ackerman, and L. A. Adamic, "Questions in, Knowledge iN? A Study of Naver's Question Answering Community," Proc. of the 27th international conference on Human factors in computing systems, 2009.
- [7] K. W. Oh and W. Bandler, "Properties of fuzzy implication operators," International Journal of Approximate Reasoning, Vol.1, No3, pp.273-285, 1987.
- [8] S. E. Robertson, S. Walker, Jones S., M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," Proc. of the 3rd Text Retrieval Conference, 1995.
- [9] E. M. Voorhees, "Overview of the TREC 2002 Question Answering Track," Proc. of the 11th Text Retrieval Conference, 2002.
- [10] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community QA," Proc. of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp.411-418, 2010.
- [11] J. W. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A Framework to Predict the Quality of Answers with Non-Textual Features," Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp.228-235, 2006.
- [12] P. Jurczyk and E. Agichtein, "Discovering authorities in question answer communities by using link analysis," Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007.
- [13] Hang Cui, Mi. Y. Kan, and T. S. Chua, "Soft

Pattern Matching Models for Definitional Question Answering," ACM Transactions on Information Systems (TOIS), Vol.25, No.2, 2007(4).

[14] K. W. Kor and T. S. Chua, "Interesting Nuggets and Their Impact on Definitional Question Answering," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp.335-342, 2007.

[15] M. Harper, D. Raban, S. Rafaei, and J. A. Konstan, "Predictors of answer quality in online Q&A sites," Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp.865-874, 2008.

[16] 안찬민, 최범기, 전석주, 이주홍, 이정식, "지식 검색 시스템에 적용 가능한 추천 질의 시스템", 한국정보교육학회논문지, 제14권, 제3호, pp.405-416, 2010.

저 자 소 개

안 찬 민(Chan-Min Ahn) 정회원



- 2003년 2월 : 인하대학교 컴퓨터 정보공학부(공학사)
- 2003년 3월 ~ 현재 : 인하대학교 정보공학부 통합과정

<관심분야> : 데이터마이닝, 웹서비스, 정보검색, SNS

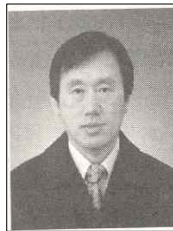
이 주 홍(Ju-Hong Lee) 정회원



- 2001년 2월 : 한국과학기술원 정보&통신공학과(공학박사)
- 2001년 ~ 현재 : 인하대학교 IT 공과대학 컴퓨터정보공학부 교수

<관심분야> : 데이터베이스, 데이터마이닝, 정보검색

최 범 기(Bumghi Choi) 정회원



- 1995년 : Florida State University Computer Science (공학석사)
- 2006년 3월 ~ 현재 : 인하대학교 정보공학부 박사과정

<관심분야> : 데이터베이스, 데이터마이닝, 정보검색, 신경망

박 선(Sun Park) 정회원



- 2001년 2월 : 한남대학교 정보통신학과(공학석사)
- 2007년 8월 : 인하대학교 정보공학부(공학박사)
- 2010년 3월 ~ 현재 : 목포대학교 정보산업연구소

<관심분야> : 데이터마이닝, 정보검색, 데이터베이스