전자거래 시스템에서 가격지정 연산자의 인식

강 승 식

한성대학교 정보전산학부
136-792 서울특별시 성북구 삼선동2가 389
sskang@hansung.ac.kr

A Recognition of Value Identifiers in Electonic Commerce System

Kang, Seung-Shik
School of Information and Computer Engineering
Hansung University

요약

전자거래 시스템에서 상품정보에 대한 자연언어 질의 문장은 상품명과 가격의 범위를 인식하는 것이 가장 중요한 요소이다. 가격의 범위를 인식하려면 가격 어휘와 가격지정어로 이루어진 가격범위 구문에 대한 별도의 처리 방법이 요구된다. 아라비아 숫자와 수사들로 구성된 가격어휘를 인식하는 수사어절 인식 알고리즘과 구문분석기를 이용하여 상품정보를 검색하는 질의 문장으로부터 상품명에 대한 가격의 범위를 인식하는 자연언어 질의어 처리 방법을 제안한다.

1. 서론

형태소 분석을 비롯하여 한글 문서를 분석할 때 기본적으로 사전에 수록되어 있는 어휘들을 중심으로 처리를 하게 된다. 일반적인 한글 문서에도 문장부호, 영문자, 숫자를 비롯하여 다양한 어휘 유형들이 포함되고 있다. 이처럼 출현빈도가 낮은 어휘들은 대부분의 응용 분야에서 중요도 및 우선순위가 낮아지게 된다.

한국어 정보처리 시스템의 일부 응용 분야에서는 문서의 유형에 따라 별도의 처리 방법이 요구되는 경우가 있다. 그 예로서 신문기사, 특허 문서, 법률 문서는 의도적인 띄어쓰기 오류가 포함되어 있으며, 전자우편과 뉴스그룹, 채팅 문장에는 맞춤법을 무시한 용어들이 다수 포함되어 있다. 이와는 별도로 숫자와 수사가 포함된 어휘들은 일반적으로 출현빈도 및 중요도가 낮지만 응용 분야에 따라 중요한 역할을 하기도 한다[1,2].

2. 수치범위 제약구문

2.1 범위제약 연산자

가격이나 개수, 크기, 날짜, 시간 등 수치의 범위를 제약하는 구문은 다양한 유형들이 가능하다. 이러한 구문은 수치의 범위를 제약하는 어휘들을 포함하고 있다. 수치범위를 제약하는 어휘를 '범위제약 연산자(range-constraint operator)'라고 정의할 때 그 예는 아래와 같다.

가격 : "백만원대의 컴퓨터를 사고 싶다"
개수 : "컴퓨터가 열대여섯개쯤 필요하다"
크기 : "15인치에서 25인치 사이의 모니터"
날짜 : "10월 중순경에 컴퓨터를 사겠다"
시간 : "매일 2시경에 출발하는 새마을호 열차"

위 예문들의 범위제약 연산자를 함수형태로 표현하면 아래와 같다.

대(백만원)
쯤(열대여섯개)
에서_사이(15인치, 25인치)
경(10월중순)
매일_경(2시)

2.2 가격지정 연산자

범위제약 연산자는 가격, 개수, 시간 등 공통적으로 적용되는 범용 연산자와 일부 혹은 특정 수치 유형에만 적용되는 개별 연산자로 구분된다. 각 수치 유형별로 개별적인 현상들이 있으므로 본 논문에서는 전자거래 시스템에서 사용되는 가격지정 연산자를 중심으로 처리 방안을 제안한다.

자연언어 인터페이스를 지원하는 전자거래 시스템에서는 상품 목록을 검색할 때 가격의 범위를 제약하는 연산자가 사용된다. 이는 상품의 최대값과 최소값을 제한함으로써 사용자가 원하는 상품을 검색하기 위한 것이다. 자연언어 질의문 처리기는 가격지정 연산자를 인식하여 그 의미에 따라 가격의 범위를 수량화하여야 한다.

- 대 : 백만원대의 컴퓨터
- 이상 : 백만원 이상인 컴퓨터
- 이하 : 3백만원 이하의 컴퓨터
- 초과 : 백만원을 초과하는 컴퓨터
- 미만 : 백만원 미만의 컴퓨터

- 에서...사이 : 90만원에서 100만원 사이의 컴퓨터
- 부터...까지 : 90만원부터 100만원까지의 컴퓨터
- 가장/제일 싼/비싼 : 가장 비싼 컴퓨터
- 최대, 최고 : 최고 백만원, 컴퓨터의 최고 가격
- 최소, 최저 : 최소 백만원, 컴퓨터의 최소 가격
- 정도 : 백만원 정도의 컴퓨터
- 가량 : 값이 백만원 가량 하는 컴퓨터
- 약/대략/대충 : 값이 약 백만원인 컴퓨터
- 짜리 : 백만원짜리 컴퓨터를 사고 싶다.
- '-' : 1-2백만원대(백만-2백만) 컴퓨터
- '~' : 1~2백만원대 컴퓨터를 사고 싶다.
- 쯤/수준 : 백만원쯤 하는 컴퓨터
- 값이 싸다 : 값이 싸고 성능이 좋은 컴퓨터
- 보다 (더) 싼/비싼 :
- x원에서 ±y원 : 백만원에서 ±10만원
- x원에 가까운 : 백만원에 가까운 컴퓨터

3. 질의어 처리기의 구조

자연언어 질의어 처리기의 입력은 질의 문장이다. 질의문은 자연언어로 주어지며, 질의문을 분석하려면 형태소 분석과 구문분석이 필수적이다. 일반적으로 자연언어 질의문으로 연구되고 있는 데이터베이스 질의어의 경우에는 형태소 분석과 구문분석, 그리고 의미분석에 의해 질의문을 분석하여 SQL문으로 변환한다[3,4].

전자거래 시스템의 질의문도 자연언어 문장이므로 구문분석과 의미분석이 필요하다. 그러나 본 논문에서는 전자거래 시스템의 질의문에서 상품명과 가격정보만 추출하는 것으로 제한한다. 즉, 형태소 분석과 구문분석만으로 상품명과 가격을 인식하는 방법이다. 이때 상품 가격에 관한 정보는 수사 어절로 표현되므로 전자거래용 질의어 처리기에서는 수사 어절을 인식하고 처리하는 기능이 매우 중요하다.

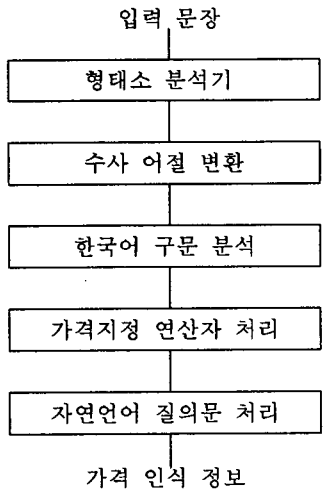

그림 1. 질의어 처리기의 구조

질의어 처리기의 구조는 그림 1과 같다. 가격어절은 한글과 숫자가 혼합되고 띄어쓰기 방법에 따라 여러 가지 유형이 가능하다. 가격어절을 인식하거나 띄어쓴 수사 어절을 표준형으로 변환하는 방법은 강승식(1999)이 제안한 방법을 이용한다. 질의문이 입력되면 전처리에서 입력 문장을 분석하여 수사 어절을 인식하고 띄어쓴 수사어절은 한 어절로 결합한다. 이 어절에 대해 형태소 분석을 통해 문법형태소를 분리하여 형태소 분석결과에 대해 수사어절을 표준형(아라비아 숫자로만 구성된 문자열)으로 변환한다.

수사어절 변환 과정을 거친 형태소 분석결과는 다시 구문분석기를 통해 구문분석 결과를 생성하고, 구문분석 결과로부터 상품명이 포함된 구문을 인식한다. 상품명이 인식된 구문에 대해 상품명과 상품명에 대한 최대-최소 가격을 구한다.

가격 어절은 일정한 패턴에 의해 인식되므로 구문분석 결과 대신에 형태소 분석 결과만을 이용하더라도 일정한 수준의 질의문 처리가 가능할 수 있다. 그러나 수식관계에 의한 상품명 인식 등 구문적 요소를 파악해야 하는 경우가 있으므로 구문분석기를 이용하는 것이 바람직하다.

4. 가격지정 연산자

수사 어절과 함께 사용되는 가격지정 연산자에는 여러 가지 유형들이 있는데, 연산자의 특성 및 출현 위치에 따라 표 1과 분류된다. '가격 어절' 유형은 '백만원짜리'와 같이 가격과 가격지정 연산자가 동일한 어절에 나타나는 경우이다. '우측 어절'과 '좌측 어절'은 가격 어절을 중심으로 가격지정 연산자의 위치를 나타낸다.

'기타'에 속하는 것으로 '가장 값이 싼/비싼'과 같은 경우는 가격 어절이 출현하지 않는다. 이 경우는 값이 가장 싼 일정한 개수의 상품을 제시할 수 있도록 가격의 범위와 별도로 focus 항목에 최대값 중심인지, 최소값 중심인지를 그 유형을 표시해 준다.

표 1. 가격지정 연산자 유형

	가격 지정 연산자
가격 어절	대, 짜리, '-', '~'
우측 어절	이상, 이하, 초과, 미만, 정도, 가량
좌측 어절	약, 최대, 최고, 최소, 최저
2어절이상	부터(에서)-까지(사이/이하/미만)
기 타	제일/가장/비교적/정말 (값)싼/비싼

가격지정 연산자는 다양한 표현이 가능하기 때문에 각각의 경우에 그 의미를 파악해야 한다. 어절 표현의 의미를 파악하려면 의미분석이 수반되어야 하나 의미분석 기술은 현재 실용적인 목적으로 활용하기가 어렵다. 궁극적으로는 가격지정 연산자의 의미분석이 수반

되어야 하는 문제이지만, 이 경우에는 그 처리 범위가 넓지 않으므로 각각의 경우에 대해 부분적인 구문분석 기술을 응용함으로써 인식이 가능하다.

가격지정 연산자의 의미는 유형에 따라 다르고 각 연산자에 대한 최대-최소값을 구하는 방법도 개별적인 의미에 따라 결정된다. 예를 들어, '1백만원대'의 의미는 1,000,000원 이상 2,000,000원 미만을 의미하고, '백만원 이하'는 0~1,000,000을 의미한다.

질의어 분석 문제의 핵심은 질의문으로부터 상품명과 가격 지정어, 수사 어절의 인식이다. 가격 지정어는 수사 어절 앞뒤에 결합되는 것이 일반적이므로 수사 어절 전후 어절을 검사함으로써 가능하다. 가격 지정어가 인식되었으면, 가격지정 연산자를 추출하여 각 연산자들에 대해 가격의 범위를 최소값과 최대값으로 변환한다.

5. 질의어 분석 및 결과 생성

5.1 출력 구조체

자연언어 질의어에 관한 연구는 데이터베이스의 사용자 인터페이스가 대표적이다. 따라서 질의문 분석결과는 SQL문 형태로 하는 것이 보편적이다. 그러나 본 연구에서 자연언어 질의어는 전자거래 시스템에서 상품을 구매할 때 사용되는 사용자 인터페이스이므로 상품명과 가격의 범위를 중심으로 정의한다.

상품구매 시에 필요한 정보는 '상품명'과 '최소값', 그리고 '최대값'이다. 이밖에도 최소값과 최대값 중에서 어떤 쪽에 더 우선 순위가 있는지를 나타내는 정보가 필요하다. 그림 2는 질의문에 대한 분석 결과를 반환하는 함수의 출력 구조이다.

```
#define MIN_PRICE      0
#define MAX_PRICE      99999999

typedef struct Query_List {
    char *prod_id;      /* 상품명 */
    int min_price;      /* 최소값 */
    int max_price;      /* 최대값 */
    int focus;          /* 가격유형 */
    struct Query_List *next;
} QUERY_LIST, *QUERY_PLIST;
```

그림 2. 질의어 처리기 출력 구조체

질의어 처리기의 출력 구조에서 최소값과 최대값이 주로 사용되지만, '100만원 정도의 값이 싸고 좋은 컴퓨터'처럼 최소값과 최대값 정보만으로는 부족하다. 이 경우에는 최대값과 최소값이 주어지더라도 100만원에 근접한 컴퓨터가 우선적으로 제시되어야 한다. 상품 목록을 순서화하는데 필요한 가격유형 정보를 표현하기 위해 연산자 유형에 따라 focus 항목에 정보를 저장한다. 질의어에 2개 이상의 상품명이 나타난 경우는 next 포인터를 이용하여 연결된다.

5.2 가격지정 연산자 인식

구문분석 결과로부터 질의어를 분석하고 그 결과를 생성하는 과정은 구문분석 트리를 탐색하면서 상품명과 가격지정 연산자를 인식하는 작업이다. 따라서 질의어 처리기의 입력은 구문분석 트리이고, 출력은 상품명과 상품가격 정보를 포함하는 구조체 리스트이다.

질의어 처리 알고리즘은 그림 3과 같다. 이 알고리즘에서 입력은 구문분석 트리의 root 포인터이다. 이 트리를 탐색하면서 각 노드마다 질의노드(query node)인지를 검사한다. 질의노드는 상품명에 대한 노드로서 상품명이 인식되면 이 노드와 자식노드들을 탐색하여 필요한 정보를 추출한다.

```
Algorithm gen_query_list(node, head)
KET_PHEAD node; QUERY_PLIST node;
{
   QUERY_PLIST tail = head;

   tail = last query node of 'node';
   if (is_prod_node(node)) { /* 상품명 노드 */
      if (head == NULL)       /* first node */
          tail = head;
      else tail = tail->next;
      set_prod_info(tail, node);/* 질의정보 */
   }

   for each children node
      head = gen_query_list(child_node, head);

   return head;
}
```

그림 3. 질의어 처리 알고리즘

그림 3에서 어떤 노드가 질의노드인지 아닌지를 검사하는 방법은 자식노드(혹은 자식노드의 자식노드) 중에서 가격 연산자와 수사 어절이 있는지를 검사하는 방법을 취한다. 상품명과 가격의 범위를 추출하는 set_prod_info()는 각 가격 연산자들의 유형에 따라 구문분석 트리를 탐색하여 최대값, 최소값을 추출한다.

가격 정보는 각 연산자의 유형에 따라 약간의 차이가 있다. 특히, 전자거래 시스템에서 실용화할 때는 이러한 정보들이 필요할 수도 있다. 가격 연산자들의 유형을 정의하면 그림 4와 같다.

```
#define QTYPE_UNKNOWN     0
#define QTYPE_ISANG       11
#define QTYPE_IHA         12
#define QTYPE_MIMAN       13
#define QTYPE_CHOKWA      14

#define QTYPE_JJEUM       21
#define QTYPE_JUNGDO      22
#define QTYPE_JJARI       23
#define QTYPE_WONDAE      24

#define QTYPE_CHOIKO      31
#define QTYPE_CHOISO      32

#define QTYPE_ISANG_IHA 41
```

그림 4. 가격 연산자 유형 예

6. 결론

전자거래 시스템에서 향후 사용자 인터페이스로서 자연언어 질의어를 처리하는 시제품을 구현하였다. 자연언어 질의어는 그 유형이 매우 다양하여 모든 유형을 포괄적으로 처리하기는 쉽지 않다. 질의문장에 대한 구문분석 결과로부터 가격지정 연산자와 가격 표시 구문을 인식하여 질의어로부터 상품명과 최소값, 최대값을 추출하였다. 가격지정 연산자의 분석 및 최대값, 최소값을 추출하는 방법은 가격, 시간, 날짜 등 수량이나 단위가 중요한 의미를 갖는 응용 분야에서 활용될 수 있다.

한국어 구문분석 기술은 일반적인 문장들을 대상으로 범용성을 지향함으로써 구조적 모호성 문제와 정확도 때문에 실용적으로 활용하기가 쉽지 않다. 본 논문에서는 전자거래 시스템에서 자연언어 질의문을 처리하기 위해 구문분석 기술을 적용하였으며, 문장 유형이 제한된 응용 시스템에서 구문분석 기술을 활용할 수 있음을 확인하였다.

참고문헌

[1] 강승식, "한국어 수사어절의 유형 분류 및 정규화", 정보과학회 추계 학술발표 논문집, 1999.

[2] 김민정, 권혁철, "한국어 형태소 분석에서의 수사 처리", 제3회 한글 및 한국어 정보처리 논문집, pp.178-187, 1991.

[3] 윤성희, "한국어 자연언어 질의 문장 파싱에서의 중의성 해소", 정보과학회 논문지(B), 24권, 12호, pp.1482-1492, 1997.

[4] 채진석, 김성기, 이석호, "한국어 데이터베이스 검색을 위한 질의 시스템의 설계 및 구현", 정보과학회 논문지, 20권, 6호, pp.810-820, 1993.

부록. 상품정보 검색 질의문 분석 예

query: 최소 1000000원의 TV를 사고 싶어요.

```
싶[F]                          싶/V 어요/e
  사[V]                        사/V 고/e
    TV[O]                      TV/N 을/j
      1000000원[G]             1000000원/N 의/j
        최소[N]                최소/N
PROD = [TV] : 1000000 ~ 999999999
```

query: 100만원 이상 150만원 이하의 TV를 보여 주세요.

```
주[F]                          주/V 세요/e
  보이[V]                      보이/V 어/e
    TV[O]                      TV/N 을/j
      이하[G]                  이하/N 의/j
        1500000원[N]           150만원/N
          이상[N]              이상/N
            1000000원[N]       100만원/N
PROD = [TV] : 1000000 ~ 1500000
```

query: 100만 이상 150만원 미만의 TV는?

```
TV[U]                          TV/N 은/j
  미만[G]                      미만/N 의/j
    1500000원[N]               150만원/N
      이상[N]                  이상/N
        1000000[U]             100만/N
PROD = [TV] : 1000000 ~ 1499999
```

query: 백오십만원짜리 tv를 구입하려 합니다.

```
하[F]                          하/V 습니다/e
  구입[V]                      구입/N 하/t 으려/e
    tv[O]                      tv/N 을/j
      1500000원[N]             백오십만원/N 짜리/s
PROD = [tv] : 1500000 ~ 1500000
```

query: 최소한 백2십만원보다 비싸고 2백만원보다는 싼 TV를 사고 싶어요.

```
싶[F]                          싶/V 어요/e
  사[V]                        사/V 고/e
    TV[O]                      TV/N 을/j
      싸[K]                    싸/V 은/e
        2000000원[U]           2백만원/N 보다는/j
        비싸[V]                비싸/V 고/e
          1200000원[U]         백2십만원/N 보다/j
            최소한[N]          최소한/N
PROD = [TV] : 1200000 ~ 2000000
```

query: 최대 5만원짜리 TV와 최소한 3만원 이상의 냉장고를 사고 싶다.

```
싶[F]                          싶/V 다/e
  사[V]                        사/V 고/e
    냉장고[O]                  냉장고/N 을/j
      이상[G]                  이상/N 의/j
        30000원[N]             3만원/N
          최소한[N]            최소한/N
        TV[&]                  TV/N 과/j
          50000원[N]           5만원/N 짜리/s
            최대[N]            최대/N
PROD = [냉장고] : 30000 ~ 999999999
PROD = [TV] : 0 ~ 50000
```

query: 최대 5만원짜리 TV와 최소 3만원 냉장고를 사고 싶다.

```
싶[F]                          싶/V 다/e
  사[V]                        사/V 고/e
    냉장고[O]                  냉장고/N 을/j
      30000원[N]               3만원/N
        최소[N]                최소/N
      TV[&]                    TV/N 과/j
        50000원[N]             5만원/N 짜리/s
          최대[N]              최대/N
PROD = [냉장고] : 30000 ~ 999999999
PROD = [TV] : 0 ~ 50000
```