질의-응답 시스템을 위한 한국어 시간 표현의 인식 및 정규화

윤도상 이도길 정후중 임해창
고려대학교 컴퓨터학과
{dsyoon, dglee, hjchung, rim}@nlp.korea.ac.kr

The Recognition and Normalization of Korean Temporal Expression for Question-Answering System

Do-Sang Yoon Do-Gil Lee Hoo-jung Chung Hea-Chang Rim
Dept. of Computer Science and Engineering, Korea University

요 약

본 논문에서는 질의-응답 시스템의 질의에서 많이 나타나는 시간 표현을 인식하고, 인식한 시간 표현에 대해서 정규화하는 방법을 제안한다. 본 논문에서 사용하는 질의-응답 시스템의 도메인은 TV방송 스케줄, 날씨 정보이며, 이러한 도메인에서는 시간 표현이 매우 빈번하게 사용되기 때문에 질의에 나타나는 시간 표현을 정확하게 인식해서 정규화하는 것이 중요하다.
제안하는 방법은 시간 표현을 의미와 기능에 따라 분류하고 각 유형마다 적절한 인식 및 정규화 기법을 사용한다. 질의에서 시간 표현은 시간 개체명 태거, 품사 태거, 시간 파서를 사용하여 인식하고, 시간 추론기와 시간 표현 사전을 이용하여 정규화한다. TV방송 스케줄과 날씨 정보 도메인의 280개 질의에서 184개의 시간표현을 이용하여 평가한 결과, 시간 표현의 인식과 정규화는 각각 93%와 96%의 정확률, 97%와 93%의 재현율을 보였다.

1. 서 론

질의-응답 시스템은 미리 구축된 데이터를 이용해서 사용자가 원하는 질의에 대한 해답을 찾아 제공해 주는 시스템이다.[1] 본 논문에서의 질의-응답 시스템은 한국어 질의를 구조화된 질의 언어 (이하 SQL)로 변환하고, 변환된 SQL을 이용하여 데이터베이스 관리 시스템 (이하 DBMS)에서 정보를 검색 한다. 그리고 검색된 정보를 이용하여 해답이 포함된 자연어 문장을 생성하여 사용자에게 제시한다. 여기에서 시간 표현의 인식과 정규화는 입력된 질의를 SQL로 변환하는 과정에서 사용된다.

본 논문에서 사용하는 질의-응답 시스템의 도메인은 TV방송 스케줄과 날씨 정보이며, 이러한 도메인의 사용자 질의는 인명이나, 장소명, 방송 프로그램명, 방송 채널명, 시간 표현 등 다양한 개체명을 포함하고 있다.

따라서 이러한 질의의 의미를 분석하여 적절한 SQL로 바꾸기 위해서는 개체명의 인식이 필요하다.

TV방송 스케줄과 날씨 정보 도메인에서의 170개 질의 문장을 분석한 결과 표 1에서와 같이 시간 표현이 다른 개체명에 비해 매우 많이 나타났다. 따라서 개체명 중 시간 표현에 대한 연구는 질의-응답 시스템에서 매우 중요함을 알 수 있다. 하지만 시간 표현은 매우 다양하게 사용되며 어휘도 제한되어 있지 않기 때문에 정확하게 인식하고 정규화하는 것은 어려운 일이다. 예를 들면 질의 (1)은 수사가 사용되어 시간을 표현하는 경우이며, 질의 (2)는 시간을 표현하는 명사가 사용되어 시

간을 표현하는 경우이다. 또한 "어제"나 "저녁"뿐만 아니라 "광복절", "어린이 날" 등과 같은 다양한 어휘가 사용되기도 한다. 질의 (3)과 질의 (4)에서는 질의를 한 날짜가 "2004년 8월 10일"이라면, "오늘"과 "2004년 8월 10일"이 같은 날짜를 표현하는 것임을 알아내는 것 또한 어려운 일이다.

개체명	포함된 질의 수 (비율)
사람이름	10 (6%)
장소 이름	32 (19%)
방송 프로그램명	12 (7%)
방송 채널명	26 (15%)
시간 표현	108 (64%)
총 개체명수	188 (100%)
총 질의 수	170

[표 1] 수집한 질의 문장에서 개체명이 나타난 비율

"2001년 이후 태풍이 몇 번이나 왔어?" (1)
"어제저녁에 한 영화 제목이 뭐였어?" (2)
"오늘 영화 뭐해?" (3)
"2004년 8월 10일에 영화 뭐해?" (4)

시간 표현의 이러한 점들을 해결하기 위해서 시간 표현을 의미와 기능에 따라서 분류한 후, 각 유형들을 시간 개체명 태거와 품사태거, 시간 파서를 이용해서 인식하고, 시간 추론기와 시간 정규화 모듈을 이용해서 정규화 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련한 기존 연구에 대해서 살펴보고, 3장에서는 한국어 시간 표현을 의미와 기능에 따라서 분류한다. 4장에서는 전체 시스템의 구성을 살펴보고, 시간 표현을 인식하고 정규화하는 방법을 설명한 후, 몇 가지 정규화 예와 SQL로 변환하는 예를 보여준다. 5장에서는 실험 방법과 결과에 대해서 분석한 후, 마지막으로 6장에서는 결론 및 앞으로의 연구 방향을 다룬다.

2. 관련연구

지금까지 시간 표현에 대한 대부분의 연구는 시간 표현을 인식하는데 초점이 맞춰져 있었다. 하지만 조응해소나 질의-응답 시스템 등의 연구가 활발해짐에 따라 이들 연구 분야와 함께 시간 표현의 정규화도 최근 많이 연구되어 지고 있다.

연구 [2]에서는 말뭉치로부터 구축한 유한 상태 변환기 (FST: finite state transducer)[1]와 어휘 데이터를 이용하여 시간 표현을 인식하고 여러 가지 구문적 역할에 대한 중의성을 해소하기 위한 방법을 제시하였지만, 시간 표현 인식에만 초점을 맞추고 있다.

연구 [3]은 스페인어를 대상으로 시간 표현을 인식하고 정규화하며, 이를 위해 문법과 사전, 형태소 분석기와 파서를 사용한다. 하지만 시간 표현을 정규화하는데 중요한 역할을 하는 "이후"나 "이전", "다음"과 같은 의미의 시간 표현은 다루고 있지 않다.

연구 [4]에서는 자연어 질의를 중간 매개 언어인 탑-언어[2](Top Language)라는 로직(Logic) 형태로 변환하고, 다시 탑-언어를 TSQL2[3]로 변환한다. 이 연구는 시간적 조응사(temporal anaphora) 해소나, NLIDB (Natural Language Interfaces for Temporal Database)의 확장, TSQL2 코드의 최적화 등 여러 분야에서 적용할 수 있는 프레임워크(framework)를 제시하고 있지만, 시간 표현을 정규화하기 위해서는 중간 매개 언어를 이해해야하는 어려움이 있다.

3. 시간 표현의 분류

많은 단어들은 서로 유사한 의미와 기능들을 가지고 있고 시간 개체명 또한 마찬가지이다. 예를 들면, "일요일"과 "월요일"은 요일을 가리키며, 그리고 "광복절"과 "어린이 날"은 특정한 날을 가리킨다. 따라서 각각의 유형에 따라 시간 표현을 인식하고 정규화 할 수 있다. 본 논문에서는 시간 표현이 포함된 170개의 질의 문장과 한국어 사전을 분석해서 147개의 기본적인 시간 표현들을 수집하였으며 이를 다음과 같이 분류하였다.

시간 표현은 크게 시간명사범주, 수식어범주, 기능어범주로 분류한다[5]. 수식어범주와 기능어범주는 자체로써 시간을 표현하지는 못하지만 시간 표현의 앞이나 뒤에 붙어서 시간 표현의 시제나 기간과 같은 의미를 부가해 준다. 시간명사범주는 조응적 (anaphoric)범주와 비조응적 (non-anaphoric)범주로 구분한다. 비조응적범주는 시간을 의미하는 의존명사 ("년", "월", "일", "시", "분", "초")나 시간 표현 기호 ("/", ".", ",", ":")가 나타나는 경우이며, 조응적범주는 수사 없이 시간을 의미하는 명사가 나타나는 경우이다. 시간명사범주를 이렇게 분류하는 이유는 두 개의 인식 방법이 구분되기 때문이다. 조응적범주는 완전시간표현과 불완전시간표현으로 구분한다[3]. 완전시간표현은 특정날짜에 의존하지 않고 자체로 특정한 날을 가리키는 것을 말하며, 불완전시간표현은 "오늘"과 같은 특정날짜에 의존하는 표현을 의미한다. 이렇게 분류하는 이유는 두개의 정규화 방법이 구분되기 때문이다.

1) FST는 속도와 공간의 매우 효율적인 활용 때문에, 음성 처리, 패턴 매칭, 품사 태깅과 같은 시스템에 있어서 많이 사용됨

2) 탑-언어는 시간 표현이 로직(Logic)으로 표현되는 문법

3) TSQL2는 구조화된 질의 언어(SQL)에서 시간의 구간 표현 타입, 시간 기간의 차이를 계산하기 위한 키워드가 추가된 일종의 데이터베이스 언어

대분류	중분류	소분류	표현 예
시간명사범주	비조응범주		2004년, 9월, 5시 20분
	조응적범주	완전시간 표현	식목일, 어린이날
		불완전시간 표현	어제, 내일, 내년
수식어범주			이번, 다음, 지난, 올
기능어범주			이전, 이후, 부터, 까지

[표 2] 시간 표현의 형식에 의한 분류

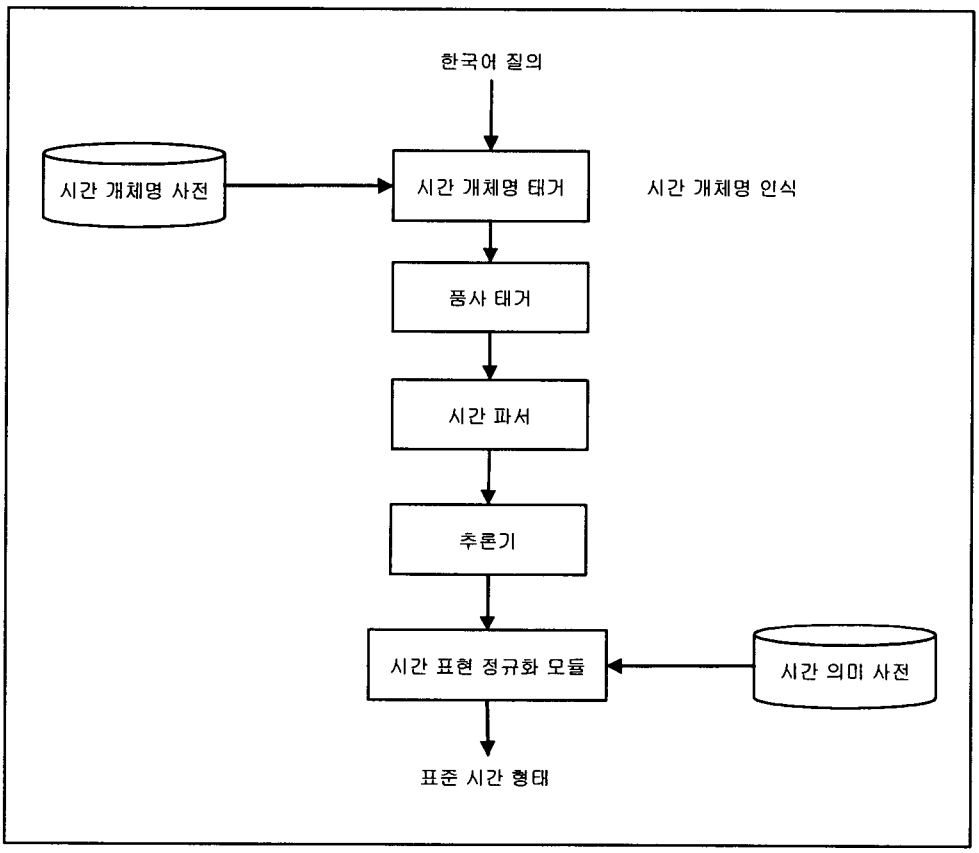
4. 시간 표현의 인식 및 정규화

4.1. 시스템 구성도

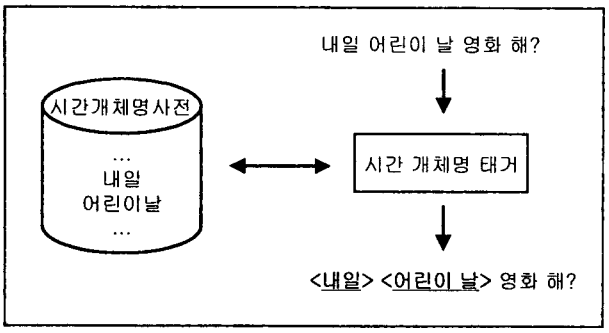
본 논문에서 제안하는 시스템은 그림 1과 같이 구성된다. 입력된 질의 문장은 시간 개체명 사전을 이용한 시간 개체명 태거에 의해서 "식목일", "어린이 날", "어제", "내일" 등과 같은 조응적범주의 유형이 인식된다. 조응적범주의 유형이 태깅된 문장은 품사태거[6]에 의해서 시간 표현 기호 ("/", ".", ",")나 시간 표현 의존명사 ("년", "월", "일", "시", "분", "초") 등을 인식할 수 있게 태깅되며, 시간 파서는 이러한 태깅을 이용하여 비조응범주, 수식어범주, 기능어범주의 표현을 인식한다. 시간 표현이 태깅된 문장은 시간 추론기를 통해서 생략된 시간 표현이 복원되며, 이 결과는 시간 정규화 모듈을 거쳐서 표준 시간 형태인 "년/월/일 시:분:초"의 형태로 정규화된다.

4.2. 시간 표현의 인식

본 논문에서는 시간 표현을 인식하기 위해서 시간명사범주를 비조응적범주와 조응적범주로 구분한다. 조응적범주인 경우, "어린이 날"과 같은 시간 표현을 형태소 분석하면 "어린이"와 "날"이 분리되어서 원래의 시간 표현의 의미가 없어져 버리기 때문에 형태소 분석 전에 그림 2와 같이 시간 개체명 사전 (147개의 시간 표현)을 이용하여 인식한다. 비조응적범주는 형태소 분석이 되면 수사와 시간 표현 의존명사 ("년", "월", "일", "시", "분", "초")나 시간 표현의 기호 ("/", ".", ",",



[그림 1] 시스템의 전체 구성도



[그림 2] 조응적 범주 인식

```
.....
(YY)YY + "/" + MM + "/" + DD
(YY)YY + "." + MM + "." + DD
(YY)YY + "," + MM + "," + DD
.....
YYYY + "년" + MM + "월" + DD + "일"
.....
HH + ":" + MM + ":" + SS
HH + "시" + MM + "분" + DD + "초"
.....
```

[그림 3] 비조응적범주를 인식하기 위한 규칙

":")가 구분되기 때문에 품사 태깅된 결과를 입력으로 받은 시간 파서는 그림 3과 같은 간단한 규칙을 이용하여 시간 표현을 인식하고, 시간 표현이 이어지는 경우는 최장일치를 적용한다. 예를 들면, "오후 9시에 영화 뭐해?"와 같은 질의에서 "오후"와 "9시"를 "오후 9시"라는 하나의 시간 표현으로 인식한다.

4.3. 시간 표현 정규화

4.3.1. 시간 표현 사전

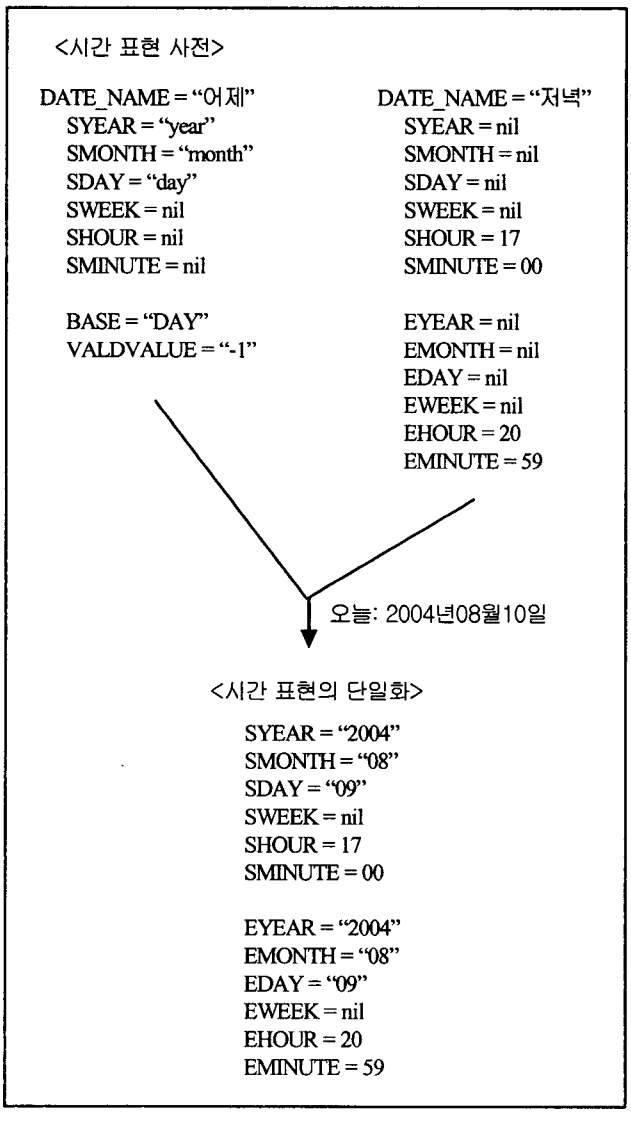
본 연구에서는 시간 개체명을 표현하기 위해서 직관적으로 알고 있는 시간 값을 이용하여 그림 4와 같이 자질-값의 구조로 구성되는 시간 표현 사전을 구축하였다. 그림 4에서 "S"로 시작하는 자질은 시작 시간을, "E"로 시작하는 자질은 종료 시간을 의미한다. 예를 들어 "어제저녁"과 같은 시간 표현은 "오늘"을 기준으로 하루를 빼서 "어제"를 표현할 수 있으므로 BASE는 "DAY"이며, VALUE는 "-1"이 된다. 그리고 "저녁"이라는 시간은 직관적으로 17시 00분부터 20시 59분으로 정의하였다. 따라서 "어제저녁"의 시간 표현은 시간 의미 사전을 이용하면 "어제"라는 시간 의미와 "저녁"이라는 시간 의미가 합쳐져서 그림 5와 같이 내부적으로 표현 값을 갖게 된다.

```
DATE_NAME = "VALUE"
            SYEAR = "VALUE"        (시작년도)
            SMONTH = "VALUE"       (시작월)
            SDAY = "VALUE"         (시작일)
            SWEEK = "VALUE"        (시작주)
            SHOUR = "VALUE"        (시작시간)
            SMINUTE = "VALUE"      (시작분)

            EYEAR = "VALUE"        (종료년도)
            EMONTH = "VALUE"       (종료월)
            EDAY = "VALUE"         (종료일)
            EWEEK = "VALUE"        (종료주)
            EHOUR = "VALUE"        (종료시간)
            EMINUTE = "VALUE"      (종료분)

            BASE = "VALUE"         (기준이되는 시간표현
                                    year|month|week|day)
            VALDVALUE = "VALUE" (기준과의 차이)
```

[그림 4] 시간 표현 사전의 자질-값 구조



[그림 5] "어제저녁"에 대한 시간 표현

```
현재시간 년도 : current_year
현재시간 월 : current_month
현재시간 일 : current_day
현재시간 시간 : current_hour

1) If (sDay ≠ 0) and (sYear = 0) and (sMonth = 0) then
   begin
      sYear = current_year
      sMonth = current_month
   end;
2) If (sMonth ≠ 0) and (sYear = 0) then
   begin
      sYear = current_year
   end;
3) If (sHour ≠ 0) and (sYear = 0) and (sMonth = 0) and  (sDay = 0) then
   begin
      sYear = current_year
      sMonth = current_month
      sDay = current_day
   end;
3) If (sMinute ≠ 0) and (sYear = 0) and (sMonth = 0) and  (sDay = 0) and (sHour = 0 ) then
   begin
      sYear = current_year
      sMonth = current_month
      sDay = current_day
      shour = current_hour
   end;
```

[그림 6] 생략된 시간 표현을 복원하기 위한 규칙

4.3.2. 시간 표현 추론기

일반적으로 사용자는 시간을 표현할 때 표준 시간 형태의 모든 정보를 다 사용하기 보다는 일부의 시간 표현을 생략시키는 경우가 많다.

"10일날 영화 뭐해?" (6)
"2004년 8월 10일날 영화 뭐해?" (7)

질의 (6)은 "년"과 "월"이 생략된 일반적인 사용자의 질의이며, 질의 (7)은 "년도"와 "월", "일"이 모두 표현된 정규화에 필요한 형태이다. 따라서 질의 (6)과 같은 경우 "2004년 8월"이라는 생략된 시간 표현을 그림 6과 같은 규칙을 이용하여 복원한다.

4.3.3. 시간 표현의 정규화

시간 정규화 모듈은 내부적으로 표현된 시간 의미를 표준 시간 형태로 변환하는 역할을 한다. 표준 시간 형태는 년도 4자리, 월 2자리, 일 2자리, 시간 2자리, 분 2자리, 초 2자리이며, 각 자리수마다 띄어쓰기로 구분을 한다. 구간을 나타내는 의미를 표현하기 위해서는 등호나 부등호를 사용해서 정규화를 한다. 이 방법은 DBMS를 사용하는 시스템에서 인식한 시간 표현을 SQL로 쉽게 변환시킬 수 있는 장점이 있다. 예를 들면 오늘이 2004년 8월 10일이라면, "어제"라는 표현은 "DATETIME = 2004 08 09"과 같이 정규화하고, "어제저녁"라는 시간 표현은 시작과 종료 구간으로 표현되기 때문에 다음과 같이 정규화 한다.

DATETIME 〉= 2004 08 09 17 00
and
DATETIME 〈= 2004 08 09 20 59

시간 표현의 수식어범주와 기능어범주는 단어 자체가 시간 개체명은 아니지만, 시간 정규화에 있어서는 매우 중요한 의미가 있다. 시간 개체명 앞에 나타나는 수식어범주는 과거나 미래 같은 시제를 의미하는 경우가 많으며, 시간 개체명 뒤에 나타나는 기능어범주의 경우는 기간을 의미하는 경우가 많다. 예를 들면, "다음 봄부터"라는 시간 표현에서 "다음"은 미래를 의미하며, "부터"는 기간을 의미한다. 본 연구에서는 시간 개체명 앞에 오는 수식어범주는 표 3과 같이, 시간 개체명의 뒤에 나타나는 기능어범주는 표 4와 같이 정규화한다. 만약 오늘이 2004년 8월이라면, "다음 봄부터"에서 "다음 봄"이라는 표현은 "2005년 3월 ~ 2005년 5월"이 되며 "부터"라는 표현은 "〉="로 정규화된다 ("DATETIME 〉= 2005 03").

아래는 현재 시간이 2004년 8월 10일이라는 가정하에서, 시간 표현을 정규화하는 몇 가지 예를 보여준다.

1. "8월 15일"
   DATETIME = 20040815

분 류	예	규 칙

지난	지난 2월	if current_month 〈= 2 then
		begin
		sYear = sYear - 1;
		sMonth = 2;
		end
		else
		sMonth = 2;

다음	다음 2월	if current_month 〉= 2 then
		begin
		sYear = sYear +1;
		sMonth = 2;
		end
		else
		sMonth = 2;

[표 3] 수식어범주의 정규화

시간 암시어	표 현
전, 이전, 까지	DATETIME 〈= "value"
후, 이후, 부터	DATETIME 〉= "value"

[표 4] 기능어범주의 정규화

2. "8월 15일 9시"
   DATETIME = 20040815 09
3. "어제"
   DATETIME = 20040809
4. "어제저녁"
   DATETIME 〉= 20040809 17
   and
   DATETImE 〈= 20040809 2059
5. "2001년 이후"
   DATETIME 〉= 2001

4.4. 구조화된 질의 언어로의 확장

SQL문은 데이터베이스를 조회하는데 사용하며, SQL문의 주어진 조건과 일치하는 데이터를 조회할 수 있다. 그림 7은 데이터베이스를 조회하기 위한 select문의 형식이다. 정규화되어진 시간 표현은 where절에서의 조건으로 사용한다. 예를 들면 "오늘 파리의 연인 몇 시에 해?"라는 질의에서 시간 표현에 해당하는 "오늘"은 사용자가 알고자 하는 방송 시간에 대한 조건이다. "오늘"이 "2004년 8월 10일"이라면, 이 질의의 시간 표현은 그림 8과 같이 where절로 표현할 수 있다.

```
Select "column1"[,"column2",etc] from "tablename"
[where "condition"];
[] = optional
      where 절에 사용되는 조건선택 연산자들
      = 같다
      > 보다 크다
      < 보다 작다
      >= 크거나 같다
      <= 작거나 같다
      <> 같지 않다
      LIKE *아래 참조
```

[그림 7] select문의 문법

```
"오늘 파리의 연인 몇 시에 하지?"

where TO_CHAR(begin_time, 'YYYYMMDD') = '20040810'
```

[그림 8] 하나의 프레임인 시간 표현의 SQL 예

```
"오늘 저녁에 파리의 연인 몇 시에 하지?"

where (TO_CHAR(begin_time, 'YYYYMMDD HH24MI')
        >= '20040905 1700')
  and (TO_CHAR(begin_time, 'YYYYMMDD HH24MI')
        <= '20040905 2400');
```

[그림 9] 두개의 프레임인 시간 표현의 SQL 예

구간으로 표현되는 시간 표현은 where절에서 "〈="와 "〉="기호를 사용한다. 예를 들면 "오늘 저녁에 파리의 연인 몇 시에 해?"라는 질의에서 "저녁"은 구간으로 표현된다. 이 질의는 그림 9와 같이 SQL문의 where절로 표현할 수 있다.

5. 실험 및 분석

제안한 방법을 평가하기 위해서 TV방송 스케줄 도메인의 280개 질의를 이용했으며, 질의에서 나타난 개체명의 구성은 표 5와 같다.

분 류	개 수	표현 예
시간 표현	184	14일, 어제, 식목일...
장르명	110	뉴스, 드라마, 영화...
프로그램 명	81	대장금, 불새...
사람이름	20	김정은, 이영애...
장소명	7	서울, 일본...
총 개체명 개수	**402**	

[표 5] 질의에 포함된 개체명의 구성

분 류		시간표현 개수	인식 및 정규화수	정답수	정확률	재현율
질의코퍼스	인식	184	191	178	93.19%	96.74%
	정규화	178	174	167	95.97%	93.82%

[표 6] 시간 표현의 인식과 정규화 결과

시간 표현 인식과 정규화에 대해서 정확률과 재현율을 구하는 실험을 하였으며, 다음과 같이 계산하였다.

◆ 시간 표현 인식

$$\text{재현율(recall)} = \frac{\text{시스템이 정확하게 인식한 시간 표현 개수}}{\text{전체 시간 표현 개수}}$$

$$\text{정확률(precesion)} = \frac{\text{시스템이 정확하게 인식한 시간 표현 개수}}{\text{시스템이 인식한 시간 표현 수}}$$

◆ 시간 표현 정규화

$$\text{재현율(recall)} = \frac{\text{시스템이 정확하게 정규화한 시간 표현 개수}}{\text{전체 시간 표현 개수}}$$

$$\text{정확률(precesion)} = \frac{\text{시스템이 정확하게 정규화한 시간 표현 개수}}{\text{시스템이 정규화한 시간 표현 개수}}$$

실험 결과는 표 6과 같다. 결과에서 볼 수 있듯이 제안한 정규화 방법은 높은 성능을 보였다. 제안한 방법은 규칙과 사전에 기반하기 때문에 정확성은 있었지만, 다른 개체명에 포함된 시간 표현도 인식해버리는 오류가 있었다. 시간 표현 인식의 오류의 예로 "이번 주 주말의 명화 제목이 뭐지?"에서 "주말의 명화"가 방송 프로그램명이지만, "주말"이라는 명사가 시간 표현으로 인식 되는 경우가 있었다. 정규화의 오류는 "추석"이나 "설"과 같은 음력의 시간 표현에 대해서 정규화를 하지 못했으며, "아침 일찍"과 같은 시간 표현에서 "일찍"에 대한 표현이 사전에 포함되지 않았기 때문에 "아침"만 정규화하는 경우가 있었다.

6. 결론 및 향후 연구

본 논문에서는 질의-응답 시스템의 질의에서 나타나는 시간 표현을 인식하고 정규화하기 위한 방법을 제시하였다. 시간 표현을 시간명사범주와 수식어범주, 기능어범주로 분류하여, 각 유형에 대해서 시간 개체명 사전을 이용한 시간 태거와 품사태거, 시간 파서를 이용하여 인식 하였으며, 자질과 값의 구조로 구축한 시간 표현 사전과 생략된 표현을 복원할 수 있는 시간 추론기, 수식어범주와 기능어범주에 따른 규칙을 이용하여 등호나 부등호의 관계로 정규화하였다. 이 정규화 방법은 시간적 조응사(Temporal Anaphora) 해소나 DBMS 데이터베이스를 이용하는 질의-응답 시스템에서 질의에 나타나는 시간 표현을 SQL로 쉽게 변환할 수 있는 장점이 있다.

향후 연구로는 질의-응답시스템의 질의에 나타나는 시간 표현만을 고려하였기 때문에, 별도의 지식이 필요한 시간 표현에 대해서는 인식 및 정규화를 할 수가 없었다. 예를 들면, "이번 아테네 올림픽 기간 동안"의 경우, "이번"과 "동안"이라는 수식어와 기능어를 이용해서 시간 표현임을 추측할 수는 있지만, 시간을 정규화하기 위해서는 추가 지식이 필요하다. 이러한 지식을 자동으로 시간 의미 사전으로 구축할 수 있는 연구가 필요하다.

7. 참고 문헌

[1] 강유환, 고병일, 서영훈, 미등록 이름 명사 인식 및 성별 구분, 제31회 정보과학회 춘계학술발표회, 31권 제1호, pp 919~921, 2004.

[2] Yoon, J., Kim, Y. K., Song, M. S. "Identifying Temporal Expression and its Syntactic Role Using FST and Lexical Data from Corpus.", Proceedings of the 18th International Conference on Computational Linguistics, pp 954~960, 2000.

[3] Estela Saquete, Patricio Martinez-Barco and Rafael Munoz, "A Grammar-Based System to Solve Temporal Expressions in Spanish Texts", Proceedings of the Third International Conference on Advances in Natural Language Processing, pp 53~62, 2002.

[4] Ion Androutsopoulos, Graeme Ritchie, Peter Thanisch, "Time, tense and aspect in natural language database interfaces", Natural Language Engineering, pp 226~276, March, 1998.

[5] 김윤관, 시간 표현에 대한 부분 문법 기술 및 FST를 이용한 시간 구문 분석에 관한 연구, 석사학위논문, 연세대학교 대학원, 컴퓨터과학과, 2000

[6] Sang-Zoo Lee, Won-Ho Ryu, Jin-Dong Kim, Hae-Chang Rim, "A Part-of-Speech Tagging Model Using Lexical Rules Based on Corpus Statistics", Proc. of the 1999 International Conference on Computer Processing of Oriental Languages, pp.385-390, 1999.