

인터넷 정보검색 인터페이스를 위한 한국어 의미분석

권혜진 노현철 이근배 이증혁
포항공과대학교 전자계산학과

Semantic analysis of Korean for web-search interface

Hyejin Kwon Hyuncheol Rho Geunbae Lee Jong-Hyeok Lee

요약

본 논문에서는 자연어를 통한 웹 정보검색 모델을 제시하고, 이를 위한 한국어 처리 방법을 소개한다. 자연어 질의에 대해 범주 문법에 기반한 구문 중심의 의미 파싱을 통해 QLF(quasi logical form)을 생성하며, 의미의 선택 제약 방법을 통해 모호성을 제거한다. QLF의 담화 처리를 통해 session 기반의 웹 검색을 실현할 수 있는 방법을 제안한다.

1. 서론

웹에서의 정보 검색이 이미 많은 사람들의 관심사가 된 것에 비해, 기존의 웹 검색 방법은 일반인이 접근하기 어려운 인터페이스를 갖고 있다. 이러한 인터페이스에서는 단일검색어나 복수개의 검색어 또는 Boolean 연산자를 이용해 질의를 하기 때문에 일반인이 자신이 원하는 정보를 효과적으로 얻기에는 다소 부자연스러운 방식이다. 이에 본 AIR-Web 시스템에서는 일반 사용자가 쉽고 편리하게 정보검색을 할 수 있도록, 세션을 기반으로한 자연어 인터페이스를 제안한다.[8]

본 자연어 인터페이스는 논리구조를 의미 표현 방식으로 사용하고, 범주 문법을 사용하는 구문 중심의 의미 분석 기법을 통해 한국어 문장을 분석한다. 이 기법은 의미 구조를 생성하는 순서와 방법이 구문 정보에 의해 제어되는 방법으로 구문 분석과 의미 분석을 결합시켜 수행함으로써 parsing과 정에서 의미 정보를 통해 불필요한 path를 줄여줄 수 있고, 체계화하기 쉬우며, 처리속도가 빠르다는 장점이 있다. 초기 단계에서 interpretation의 가능성이 확장되는 것을 방지하면서 전체적으로 시스템의 복잡도를 줄이기 위해 의미 분석의 중간단계인 QLF(Quasi Logical Form)를 도입한다. QLF 생성 후에는 국소 담화 분석(local discourse analysis)을 통해 참조 해결(reference resolution)과 생략(ellipsis)을 처리하고 최종 논리구조(Logical Form)를 생성하게 된다.[7]

2. 세션기반의 자연어 웹검색

2.1. 검색 모델

기존의 웹 검색 엔진들이 지원하는 웹 문서에 대한 질의어 구성 방식에는 단일 검색어 방식, 복수개의 검색어 조합, 불리안(Boolean) 연산자를 이용한 질의어 등이 대표적이다. 그 정보검색과정은 아래 그림 1에서 보는 바와 같이 두 단계 과정의 단순 상호작용으로 전제하고 있다.

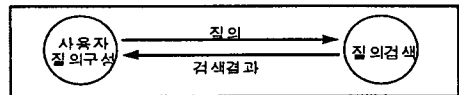


그림 1. 일반적인 검색모델

그러나, 실제 사용자의 웹 검색 과정을 살펴 보면, 단 한번의 질의 검색으로 원하는 정보를 얻는 일은 매우 드물고, 대부분 여러번 검색을 반복하면서 점차 원하는 정보에 접근하게 된다. 또한 이 과정에서 우리가 주목할 만한 점은 앞서 구성된 질의어와 그에 대한 검색 결과는 다음 질의어를 구성하는데 많은 영향을 주게 된다는 점이다. 즉, 앞선 검색 결과(질의어, 결과)를 토대로 이전의 질의어를 연속적으로 수정해 가면서 원하는 정보에 접근하는 것이 사용자 입장에서는 보

다 자연스러운 웹 검색 과정이다. 즉, 웹 검색 과정이란 아래 그림 2와 같이 3 단계가 하나의 사이클로 연속적인 질의어 재구성을 통해 원하는 정보에 접근해 가는 과정이라고 정의할 수 있다. AIR-Web에서는 사용자가 원하는 정보를 얻기까지 이러한 반복 과정을 하나의 세션(Session)으로 정의하여 통합적으로 관리한다.

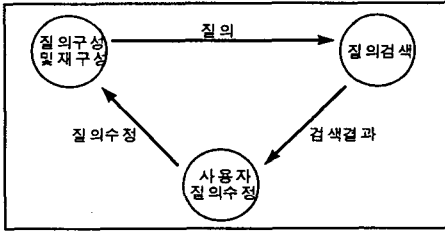


그림 2. 세션기반의 검색모델

2.2. AIR-Web 시스템

AIR-Web은 Web상에서의 정보 검색에 있어 사용자에게 보다 쉽고 편리한 환경을 제공하기 위한 목적으로 개발된 정보 검색도구로서 다음과 같은 점들로 크게 특징지을 수 있다.[8]

- Multi-Agent환경
- 메타써치(Meta Search) 엔진의 형태
- 자연어 처리 모듈
- 질의어 재구성을 지원하기 위한 질의어 관리 모듈(세션기반 질의어 관리 모듈)

이상의 특징을 가진 AIR-Web의 시스템 아키텍처를 보면 다음과 같다.

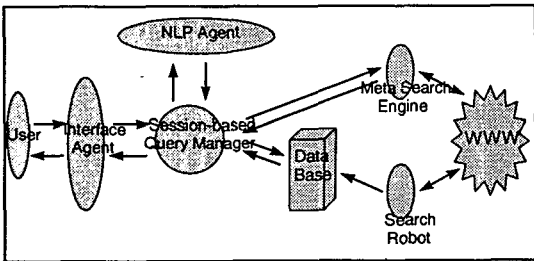


그림 3. AIR-Web 시스템의 구조

2.3. 자연어 세션의 특징

그림 4는 자연어 세션의 예를 보여준다.

Goal: 인터넷의 어느 컴퓨터에 Julia라는 채팅로봇이 있다. 사람과 말상대를 해 주는 이 로봇의 위치를 찾아라.

- ① Julia와 chat나 talk이란 단어를 포함하고 있는 문서를 찾아 줘.
- ② 그 중에서 chat이란 말이 있는 건 빼고 보여 줘.
- ③ talk대신 chat으로 바꿔 봐.
- ④ 거기에서 chatting robot과 관련된 것만 골라 봐.
- ⑤ 2의 결과에서 robot이란 말이 있는 문서만 골라 봐.
- ⑥ 그 중에서 turing test와 관계있는 것만 골라 봐.

그림 4. 자연어 세션의 예

자연어 세션은 다음과 같은 특징을 가지고 있다:

- 미등특어가 많다. 사용자가 검색하고자 하는 검색어들은 도메인과 어휘수가 제한 될 수 없기 때문에 미등특어가 많이 발생한다.
- 사용자의 질의에는 검색어뿐만이 아니라 시스템의 제어에 대한 명령이 포함되어 있다. ③에서 "talk"이라는 단어 대신에 "chat"이라는 단어로 검색어를 바꾸려면 사용자의 의도를 파악하여 처리하여야 한다.
- 대응어와 생략이 많다. 이전의 질의와 질의 결과 등은 대응어를 통해서 표현하기 쉽고, 동사 등의 생략도 자주 일어난다.
- 모호성이 있는 결과중에서 하나의 결과를 뽑아야 질의어 생성에 사용할 수 있다.

따라서 이러한 점을 충분히 고려하여 질의분석을 해야만, 사용자의 의도에 적합한 결과를 보여줄 수 있다.

3. 구문/의미 분석기의 구조

AIR-Web의 자연어 처리 모듈에서는 사용자의 질의들이 형태소 분석과 구문 분석, 의미 분석 및 담화 분석이라는 일련의 과정을 거쳐 처리되며, 결과로써 한국어 논리구조를 생성하게 된다 (그림 5).

자연어 질의는 형태소 분석기를 거쳐서 형태소 분석결과인 형태소 그래프로 바뀌고, 이 형태소 그래프에 대해 범주 문법과 논리구조에 기반한 구문/의미 분석이 이루어진다. 구문/의미분석을 위한 범주 문법 파싱은 category cancellation에 의해 bottom up방식으로 이루어 지고, 파싱과정에서 의미 결

합이 이루어진다. 각 구문 범주당 여러개의 의미 범주가 있을 수 있기 때문에, 의미 분석의 결과는 모호성을 갖게 되며, 이것을 해소하기 위한 방법으로 범주제약과 의미제약을 이용하여 가능한 결과들의 분석점수를 구하고 그것을 기반으로 가장 좋은 결과를 고른다. 이것이 끝나면, 양화사 및 논리 연산자(logical operator)에 대한 scoping 해결을 하고, 아직 문맥 내에서 의미가 해결되지 않은 대명사와 anaphoric term의 의미를 국소 담화 분석을 통해 해결함으로써 최종 의미 분석 결과를 주게 된다.[7]

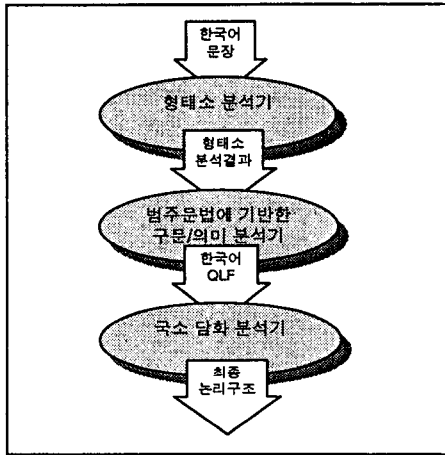


그림 5. 자연어 질의의 분석 과정

4. 한국어의 논리 구조

한국어 어휘의 기능 및 의미에 따른 논리구조는 [7]에 기반을 두고 있다. 본 논문에서는 확장 또는 변경된 구조와 특별히 웹 검색에서 중점을 두어야 할 것에 대해서만 다루기로 하였다.[1][2][10]

4.1. 명사

가) 일반명사

명사는 그것을 논리구조로 나타내는데 있어서 일반적으로 다음의 두가지 요소에 의해 그 의미가 결정된다. 첫째는 화제가 되고 있는 것이 물질이나 집합 전체인가, 아니면 그 구성 요소의 한 특례인가이고, 둘째는 한 특례일 경우 어느 특례가 화제가 되고 있는지, 청자에게 알려져 있는가, 그렇지 않은가 하는 점이다. 이것을 표현하기 위해 term이라는 구조

를 도입했으며, term 자체는 ambiguity가 해결되지 않은 상태가 된다. 다음은 term구조의 표현 방식이다.

$term(\langle type, lexical, number \rangle, variable, [noun, variable])$

type은 category type의 종류를 나타내고, lexical은 실제 쓰인 어휘를 나타내며, number는 그 어휘가 복수의 의미인지 단수의 의미인지를 나타내는데 쓰인다. 다음은 '그 사용자'에 대한 논리적 구조이다.

$term(\langle type=ref, lexical=the, number=sing \rangle, X1, [user, X1])$

이 경우 문맥내에서 '그'의 의미가 jinny1으로 해석될 수 있다면, '그 사용자'는 참조해결시 jinny1으로 대체되게 된다.

나) 공동격조사와 결합된 명사

'와나' 과와 같은 공동격 조사와 결합된 명사를 인자로 취하는 논리구조는 모호성을 내포하고 있다. 예를들어 'A문서와 B문서를 합치다'라는 문장과 'A문서와 B문서를 찾다'라는 문장을 살펴보면, 전자는 하나의 이벤트에 대한 문장이지만, 후자는 A문서와 B문서 각각에 대해 서로 다른 이벤트가 관여되어있는 문장임을 알 수 있다. 이러한 모호성은 구문파싱중에 해결하기 보다는 term_conjoin이라는 새로운 구조를 사용해서 표현하며 의미분석 이후의 스코핑단계에서 해결한다.

다음은 'A파일과 B파일'을 term_conjoin을 사용해 간단히 표현해 본 것이다.

$term_conjoin(\text{and}, C, A\text{파일}, B\text{파일})$

이 논리구조는 나중에 scoping을 처리할때 collective reading으로 처리되거나 distributive reading으로 처리된다. 위의 예문을 간단하게 논리구조로 표현하면 다음과 같다.

- A문서와 B문서를 합치다.

$[\text{combine}, EV1, SUBJ, \text{set}(A\text{문서}, B\text{문서})]$

- A문서와 B문서를 찾다.

$[\text{and}, [\text{search}, EV2, SUBJ, A\text{문서}], [\text{search}, EV3, SUBJ, B\text{문서}]]$

다) 미등록어(검색어)

웹 검색에서는 대부분의 검색어가 미등록어이기 때문에 이것에 대한 형태소 정보 및 구문, 의미 정보를 추정해야 한다. 본 시스템에서는 검색어 이외에 필요한 모든 필수 어휘가 등록되어 있다고 가정하여, 형태소 분석 단계에서 모든 미등록어를 명사로 추정하고 구문정보는 'np'를, 의미정보로는 $term(\langle def, bare, sing \rangle, X, [\text{검색어}, X])$ 를 준다.

라) 관계명사

'저자', '업마', '주인' 등과 같은 명사는 '개', '사용자' 등과의 보통 명사와는 다르게 쓰인다. 이 명사들은 어떤 개체의 형태를 의미하는 것이라기 보다는 다른 개체와의 관계로 정의된다고 할 수 있다. 따라서 이러한 명사들은 다른 명사에 의해 수식을 받는 형태로 주로 쓰이게 된다. 즉, 'A의 주인'과 같이 'A'와의 관계에 의해 정의된다고 할 수 있으며, 이것에 대한 간단한 논리적 형태는 $term(<.>, X, [and, [person, X], [own, X, A]])$ 과 같이 나타낸다. 이러한 관계명사들은 수식하고 있는 명사와의 관계를 해결하는 일이 필요하며, 그 관계는 selectional restriction과 같은 과정을 통해서 의미 검사를 통해 해결한다.

4.2. 용언

형용사와 동사는 논리적 구조에서 인자(argument)로 취하는 필수 성분의 수에 따라 n항 술어(n-place predicate)로 구분된다. 예를 들어 '옮기다'의 경우 two-place predicate으로 주어와 목적어를 인자(argument)로 취하며, 그 기본 구조는 다음과 같다.

$[move, EV, _ : np[j]; N1; _ : np[j]; N32]$

이것의 의미는 move가 취하는 두번째 argument는 np[j]의 문법 범주로서, 사람(N1)이어야한다는 것을 의미하고, 세번째 argument는 np[j]의 문법 범주로서, 사람이 아닌 physical object(N32)이어야 함을 나타낸다. 필수 성분 이외의 성분은 수식어구로 처리하며, 주로 event에 대한 predicate으로 나타낸다.

4.3. 부사

부사는 용언(서술어)를 수식하는 품사로서, 형태적으로 고유한 부사와 형용사 어간에 '이', '히', '게' 등이 붙은 파생부사로 나뉜다. 부사는 의미분석시 서술어의 event에 대한 술어로서 기술되며, 용언과 같은 event 변수를 공유하게 함으로써 그 event에 대한 부연 설명을 하게 된다. 이것은 부사격조사의 처리 방법과 같으며, 예를 들어 '빨리 찾아'에 대한 논리구조는 다음과 같다.

$[and, [search, EV1, SUBJ, OBJ], [quick, EV1]]$

4.4. 조사

조사는 논리적 구조를 표현하기 위해 문법 기능을 담당하는 조사와 담화 기능을 담당하는 조사, 공동격 조사, 부사격 조사, 관형격 조사, 비교격 조사등으로 나뉜다.

가) 문법기능조사

문법 기능을 담당하는 조사중 주격조사(이/가, 게/서)와 목적격 조사(을/를) 등의 경우 파싱 과정에서 결정되기 때문에 의미 정보를 가질 필요가 없다. 즉 '사용자가'라는 어절에서 '사용자'와 '가'가 결합될때는 범주로서 np[j]가 남겨지며, 이것은 범주제약을 검사하게 될때 정보로서 활용된다.

나) 관형격(속격)

관형격 조사는 논리적 구조에서 서술(predicate)로 나타나게 되며, 그 내포적 의미를 해결해야 한다. 관형격 조사에 의해 수식받는 명사가 관계명사인 경우, 관계에 대한 서술(predicate)은 사전에 등록되어져 있어야 한다. 예를 들어, "이 책의 저자"에서 관계 명사인 "저자"가 사전에 등록된 형태는 다음과 같다

$term(<ref, bare, sing>, X, [and, [person, X], [author_of, X, _ : np[np; N사물]])$

그 외의 명사가 관형격 조사에 의해 수식받는 경우는 소유관계로 해석이 가능하다. 예를 들어, "지니의 책"에 대한 논리구조는 다음과 같이 나타낼 수 있다.

$term(<ref, bare, sing>, X, [and, [book, X], [own, jiny, X]])$

'의'는 소유격 조사외에도 동격의 의미로 쓰인다. '각각의 파일', '열명의 사람'에서 쓰인 '의'가 바로 동격의 '의'이다. 이에 대한 간단한 논리구조의 예는 다음과 같다.

$[equal, 열명, 사람]$

다) 공동격 조사

앞에서 언급된 바와 같이 처리되는데, 예를 들어 'A나B'와 같은 어절을 간단히 표현하면, $term_conjoin(operator=OR, variable=C, A, B)$ 로 처리하게 된다.

라) 부사격조사

의미	종류	논리구조
장소	에	$[and, /_ : s\$X, [in_loc, EV, _ : np; N장소]]$
시간	에	$[and, /_ : s\$X, [at_time, EV, _ : np; N추상시간]]$
대상	에,에게	$[and, /_ : s\$X, [to_person, EV, _ : np; N사람]]$

표 1. 부사격조사와 논리구조

부사격 조사는 부사어구를 만들며, 서술어에 대한 수식어로 쓰이게 된다. 부사격 조사에 의해 형성된 부사어에 대한 논리적 구조는 이미 앞에서 기술한 바와 같다. 부사격 조사중

'와', '과'와 같은 공동격조사와 '보다'와 같은 비교격조사는 그 의미의 특성에 의해 다른 방법으로 처리하며, 그 외의 부사격 조사는 하나의 predicate처럼 논리구조에서 나타나게 된다. 표 1은 부사격조사에 대한 논리구조의 예이다.

5. 규칙을 이용한 확장

관형형어미 '는', '니', '은' 등과 관형격조사 '의'가 쓰인 명사구는 결합과정에서 합성성(compositionality)을 보장하지 못한다.

예를 들어 'Julia를 포함하는 문서'라는 관형절을 살펴보기로 하자. 'Julia를 포함하'에 대해 생성된 논리구조를 간단히 나타내면 $[include, EV, _np[가], Julia]$ 이고, '문서'에 대한 논리구조는 $term(<quant, bare, sing>, X, [document, X])$ 이다. 이 두 논리구조를 '는' 에 의해 결합시키면

$$term(<quant, bare, sing>, X, [and, [document, X], [include, EV, X, Julia]])$$

와 같이 되어야 하며, 이 과정에서 관형형어미 '는'은 앞의 관형절의 논리구조가 뒤의 term에 부가적 의미를 부여하면서, 뒤의 명사가 앞의 관형절의 slot를 채우는 과정이 필요하게 된다. 즉 이러한 구조에선 합성성의 원리가 유지되지 못하므로 이러한 것을 처리하기 위한 규칙의 도입이 필요하다. 위에서 언급한 '는'의 경우 '는'의 앞에서 결합되는 관형절이 '는'뒤의 term에 결합되면서 앞의 관형절의 비어있는 slot을 뒤의 명사가 채워지는 것에 대한 규칙을 사용하여 해결한다.

6. 범주 문법에 기반한 의미 합성

6.1. 범주 문법

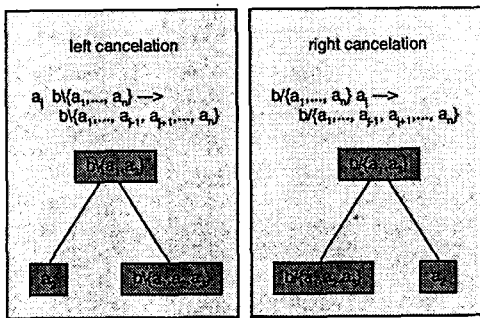


그림 6. category cancellation

범주 문법을 사용하여 한국어를 모델링하게 되면, 구구조 문법과는 달리 한국어가 가지는 후치사와 자유어순을 쉽게 처리할 수 있게 된다. 범주 문법은 구문 규칙을 사용하지 않고, 사전에 정의된 범주(category)에 따라 bottom-up방식으로 category cancellation에 기초해 구문 분석이 이루어 지는 파싱 방법이다.[5]

그림 6은 한국어의 자유어순을 처리하기 위한 category cancellation의 방법이다.

6.2. 의미 합성 과정

의미합성 과정에서는 범주 제약과 의미 제약을 사용하여 두 개의 논리구조의 결합 가능성을 검사하고, 결합된 논리구조의 preference를 계산한다.

범주제약은 slot에 채워질 filler의 구문범주에 대한 제약을 두어 타당한 문법 범주를 지닌 논리구조만이 오게하는 것이고, 의미제약은 범주제약과 비슷한 방법으로 slot에 의미제약을 두는 것이다. 이때 compositional하게 분석하기 힘든 부분을 위해 앞에서 정의한 규칙을 사용한다.[6][7]

6.2.1. 기본적인 합성방식에 의한 결합

범주 문법에 기초한 의미분석은 두개의 구문 범주간에 category cancellation이 적용될 때마다 의미의 합성이 이루어지게 한다. 결합될 두개의 의미 카테고리(semantic category)는 대체로 functor와 argument의 관계로 되어 있으며, argument의 구문 및 의미 정보가 functor의 slot에 가장 적합할때 가장 높은 점수를 주게 된다. 두개의 논리구조가 결합될때에는 먼저 functor의 slot에 있는 구문정보와 argument의 구문정보가 일치하는지를 확인한 후 의미의 합성 단계로 넘어가게 된다. 그림 7은 구문 분석 과정에서 의미의 합성이 이루어지는 방식을 보여주고 있다.

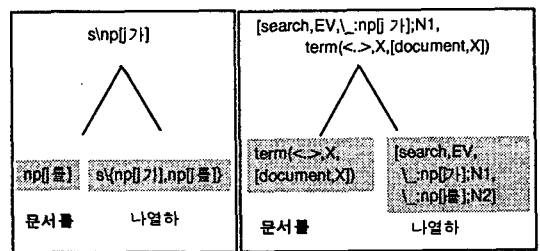


그림 7. 의미 합성 과정

다) 의미적 유사도의 계산

의미 제약은 앞에서 언급한 바와 같이 술어가 취하는 인자(argument)가 의미적으로 타당할때 결합되도록 제약을 가하는 방식이다.

'문서를'과 '나열해'의 논리구조가 결합될때 '문서를'은 의미정보로서 {impersonal(N2), abstract(N4), spatial(N5), nonrelational(N7)}을 갖고 있고, '나열해'의 두번째 slot은 채워질 논리 구조의 의미 정보로서 {impersonal(N2)}을 요구한다. 따라서 결합이 타당하기 때문에 '문서를'의 논리구조가 '나열해'의 두번째 slot을 채우게 된다. 이러한 의미 제약은 두 논리구조가 결합될때마다 아래의 유사도에 따른 점수를 계산하여 파싱이 끝났을때 각 파싱결과마다 최종 점수를 갖게 되며, 이중 가장 점수가 높은 것을 결과로써 선택하게 된다.

argument의 논리구조와 functor의 논리구조가 결합될때 결과로 생기는 구조의 점수는 다음과 같다.

$$\text{result score} = C1 \times \text{matching score} \times \text{argument score} + (1-C1) \times \text{functor score}$$

$$\text{matching score} = \frac{2 \times |A \cap B|}{|A + B|}$$

matching score는 argument와 functor의 slot이 결합될때 argument의 semantic category와 functor에 있는 slot의 semantic category가 어느정도로 matching이 되는지를 계산한다. 결합된 논리구조는 functor와 argument가 가지고 있는 점수와 matching score를 가지고 자신의 점수를 계산하게 된다. C1은 상수로서 실험치이다.[9][11]

7. 스코핑과 담화처리

7.1. 스코핑처리

scoping ambiguity는 한국어에 있어서 빈번하게 나타나는 현상은 아니지만, quantifier와 여러 논리연산자 등에서 일어난다. quantifier와 operator의 scope를 결정하는 것은 형태소, 구문, 의미 정보 외에도 문맥 정보에 의해 많이 영향을 받는다. 그러나 문맥을 고려하지 않더라도 몇가지의 제약과 선호도에 의해서 결정될 수 있다.

먼저 양화사(quantifier)와 그에 의해 생기는 모호성을 예를 들어 설명하기로 하겠다. '누가 모든 기계에 로그인해 있지?'라는 문장과 '각각의 기계에 누가 로그인해 있지?'라는 문장을 비교해 보기로 하자. 이 두 문장은 모두 scoping ambiguity를 갖고 있지만 전자의 경우 '모든'이 '누가'를

outscope하는 것으로 읽혀지는 경향이 있고, 후자의 경우 '각각'이 '누가'를 outscope하는 경향이 있다.

한국어의 논리구조에서 양화사인 경우 type이 quant로 정해지며, lexical의 정보에 따라 scoping의 순서가 결정된다. 이미 앞에서 scope ambiguity를 담고 있는 term의 구조에 대해 살펴본 바와 같이, term의 구조는 <category,variable,body>로 이루어져 있고, category는 <type,lexical,number>의 정보로 이루어져 있다. type이 quant인 경우에는 quantifier에 대한 scoping이 필요하다는 것을 나타내고, type이 ref인 경우에는 우선 문맥내에서의 의미파악을 한후, 그것이 불가능할 경우 quantified expression으로 간주한다.[1][3]

한국어의 양화사및 logical operator에 대한 일반적인 양화사(quantifier) strength의 계층은 다음과 같다.

시제연산자 > 각각의 > 의문(양화사) >
모든 > 한,두 > 부정연산자

7.2. 국소 담화 분석

국소 담화 분석과정에서 가장 핵심이 되는 참조해결과 생략 처리의 경우, 본 시스템에서 이용되는 담화는 웹검색이라는 한정된 도메인이기 때문에 비교적 쉽게 처리할 수 있다. 참조 중심(center, Discourse focus)는 항상 '검색된 문서의 집합'이거나, 검색된 문서 집합의 개체인 문서, . Session history에서 선택된 질의어등으로 한정되어 있어서 약간의 구문 정보와 의미 정보를 사용하여 참조해결이 가능하다.[1][8]

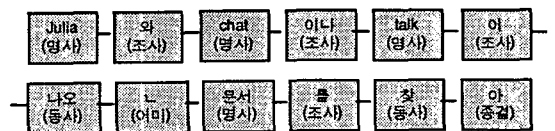
8. 실행예제및 결론

8.1. 실행예제

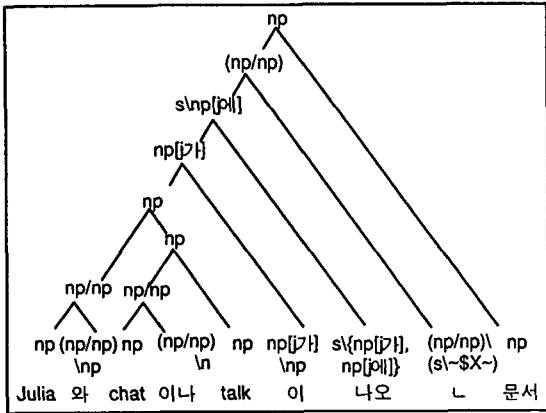
다음은 자연어 질의가 형태소 분석과 구문 분석, 의미 분석 및 담화분석이라는 일련의 과정을 거쳐 처리되는 결과를 나타내고 있다.

- 질의 : Julia와 chat이나 talk이 나온 문서를 찾아.

가) 형태소 분석 결과



나) 구문분석결과



선의 결과를 내고 있으며, 국소 담화 분석을 통해 연속된 질의어가 한 session내에서 체계적으로 처리되도록 하였다.

9. 참고문헌

- [1] Allen, J. 1995. Natural Language Understanding, The Benjamin/Cummings Publishing Company, INC
- [2] Alshawi, H., and van Eijck, J. 1989. Logical forms in the core language engine. In Proceedings of the 27th Annual Meeting of the ACL.
- [3] Alshawi, H. 1990. Resolving quasi logical forms. Computational linguistics 16(3).
- [4] Covington, M. A. 1994. Natural Language Processing For Prolog Programmers. Prentice Hall.
- [5] Lee, W. 1995. Chart-driven connectionist categorial parsing of spoken Korean. In Proceedings of the 1995 International Conference on Computer Processing of Oriental Languages.
- [6] Moore, R.C. 1989 Unification-based semantic interpretation. In Proceedings of the 27th Annual Meeting of the ACL.
- [7] 권혜진의 3인 1996. 범주문법에 기반한 한국어의 의미분석. 한국정보과학회 23-1 학술발표논문집
- [8] 노현철의 3인 1996. 한국어 담화 분석을 이용한 session 기반의 웹 검색 시스템 : AIR-Web. 한국정보과학회 23-2 학술발표논문집
- [9] 김은자. 1994. 언어 패턴에 기반한 일-한 기계 번역 시스템. 포항공대, 전산학과, 석사논문.
- [10] 권재일. 1994. 한국어 통사론. 민음사.
- [11] 寺村秀夫. 1990. 일본어의 구문과 의미. 범문사.

다) 의미분석 결과

```
[imp, [search, EV1, \_np[?i],
term(<quant, bare, sing>, X4,
[and, [document, X4],
[contained, EV2,
term_conjoin(and, C2
term(<def, bare, sing>, X1, [named_string, X1, "Julia"]),
term_conjoin(or, C1,
term(<def, bare, sing>, X2, [named_string, X2, "Chat"]),
term(<def, bare, sing>, X3, [named_string, X3, "talk"])),
X4]]]]]
```

나) 스코핑처리 및 담화분석 결과

- Julia와 chat이나 talk이 나온 문서를 찾아.

```
[imp,
[quant(exists, X4,
[and, [and, [document, X4],
[contained, EV1, term(<, -, Julia), X4]],
[or, [and, [document, X4],
[contained, EV1, term(<, -, chat), X4]],
[and, [document, X4],
[contained, EV1, term(<, -, talk), X4]]]]],
[search, EV1, \_np[?i], X4]]]
```

- 그중에서 chat이 들어있지 않은 것만 보여줘.

```
[imp,
[quant(exists, X5, [and, [and, [thing, X5]
[부정, [contained, term(<, -, chat), X5]],
[among, X5, query(prev)],
[봉사, [show, EV2, \_np[?i], X5]]]]]]]
```

8.2. 결론

본 논문에서는 한 session내에서 계속 수정되는 자연어 질의어를 통한 웹 검색 모델을 위한 한국어의 의미분석기를 설계, 구현하였다. 한국어의 품사 및 의미에 맞게 논리적 구조를 정의하였고, 범주 문법에 기반한 구문 분석 과정에서 의미분석이 이루어지게 하였다. 모호성의 해결을 위해 구문제약과 의미제약을 두어 각각의 문장마다 score를 계산하여 가장 최