문장 표현 언어 VAR(Verb Activate to Relation) 설계

○
김 경 서, 송 만 석
연세 대학교 전산과학과

Design of Sentence Representation Language VAR (Verb Activate to Relation)

Kyeongseo Kim, Manseok Song
Department of Computer Science, Yonsei University

요 약

컴퓨터가 자연 언어를 이해하기 위해 가장 필요로 하는 것은 단어에 대한 정보다. 단어는 문장 안에서 나름대로의 정보를 지니고 사용된다. 사람들이 실제 생활에서 사용하는 문장을 대량으로 모아 둔 것을 말뭉치(Corpus)라 한다. 그러므로, 단어가 인간 언어 생활에서 사용되면서 지니는 정보를 찾기 위해서는 말뭉치를 들여다 보는 것이 필요하다. 본 논문에서는 문장이 갖고 있는 정보 중 많은 것을 표현할 수 있는 언어, VAR를 설계한다. 그리고 말뭉치를 VAR로 표현해서 관리하면서 언어학자 및 전산학자가 좋은 지식 기반(Knowledge Base)를 만들 수 있는 기초를 제공한다.

1. 서 론

대개의 자연언어 처리 시스템은 처리의 기본 단위로 문장을 삼는다. 문장 분석은 어떤 단어들로 문장이 이루어져 그 모양이 어떠하며, 단어들이 어떻게 문장의 구조를 만들어 그 구조는 어떠하고, 어떻게 문장의 의미를 이루어서 그 의미가 어떠한가를 아는 것이다. 그러므로, 컴퓨터가 문장을 분석하기 위해서는 단어가 문장에 어떻게 나타나며(형태 정보), 문장 구조를 이룰 때 어떤 작용을 하고(구문 정보), 문장의 의미를 나타내는 데 어떤 역할을 하는지(의미 정보)를 알고 있어야 한다.

단어는 문장에서 사용된다. 또 단어는 여러 문장에서 제각기 나름대로의 정보를 갖고 다양하게 사용된다. 그러므로 단어의 정보를 정확하고도 범용적으로 알기 위해서는 많은 문장들을 살펴 보아야 한다. 대량의 문장들을 과학적인 방법으로 모아둔 것을 말뭉치라 한다. 그러나, 문장 형태로 이루어진 말뭉치(단순 말뭉치)에서는 체계적으로 단어에 대한 정보를 찾기는 매우 힘들다. 말뭉치가 효과적으로 이용되기 위해서는 단어의 정보를 잘 찾을 수 있도록 구조화되어야 한다.

본 논문에서는 한 문장이 가지고 있는 정보를 효과적으로 나타낼 수 있는 언어, VAR를 설계한다. 그리고 VAR를 이용해서 단순 말뭉치를 구조화한다. 단순 말뭉치에서는

같은 단어라 할지라도 다른 문장에서 사용되면 서로 떨어져 있어 이 단어에 대한 정보를 찾기 위해서 다른 방법을 사용해야 한다. 또 단어의 의미나 기능이 다르게 사용된 경우에도 이를 표시할 방법이 없기 때문에 단어의 의미 정보를 일관성 있게 추출할 수 없다. 그러나, VAR로 구조화된 말뭉치에서는 단어가 전체 말뭉치 문장에서 나타나는 경우마다 연결된다. 또, 단어의 쓰임새에 따라 다양하게 연결될 수 있어 단순 말뭉치보다 많은 정보와 일관성 있는 정보를 제공할 수 있다.

또, VAR로 표현된 말뭉치는 연결고리를 일반화하고 다양하게 하는 과정에서 자연스럽게 실세계의 정보를 지니는 지식 기반(Knowledge Base)으로 발전할 수 있다.

2. VAR 소개

대부분의 의미 표현 언어는 동사를 문장의 중심어로 취급한다. 동사가 요구하는 논항(argument)에는 어떤 것들이 있으며, 문장에서 각 논항에 어떤 명사가 할당되는가를 표현한다. 어떤 의미 표현 언어는 시간이나 장소 등의 정보를 나타내기도 한다. 그러나, VAR는 의미 표현에서 다시 문장을 생성해야 하기 때문에 의미 표현 언어보다 구체적인 정보를 가질 수 있어야 한다. 여러 문장들이 공통된 단어를 가지고 있으면 서로 연결하고 단어들이 어떤 관계를 가지면 이들 사이의 관계를 나타낼 수 있어야 하기 때문에 지식 표현 언어와 비슷한 점도 갖고 있다. 문장의 의미를 나타내면서 문장 구조에 대한 정보도 같이 가지는 언어를 문장 표현 언어라 정의하면, VAR를 문장 표현 언어라 할 수 있다. VAR는 대체적으로 망(network)구조를 가진다. VAR는 몇 가지 종류의 노드(node)와 이들 노드들을 연결해 주는 링크(link)로 구성되어 있다. 모든 링크에는 링크의 의미를 나타내는 꼬리표(lable)가 있다.

2.1 Relation 도입

VAR의 노드는 기본적으로 명사 노드, 동사 노드 그리고 Relation 노드로 이루어진다. 명사 노드는 "생각", "학교", "음식" 등과 같이 일반적으로 명사가 나타내는 개념을 나타낸다. 동사 노드는 "먹다", "예쁘다" 등과 같은 동사, 형용사가 가지는 개념을 나타낸다. Relation 노드는 동사 노드로부터 활성되는 것으로 정의한다.

활성화(activation)는 동사가 구체적인 환경에서 사용될 때 동사가 구체적인 의미를 가지게 되는 것을 말한다. 동사가 사용되는 구체적인 환경은 동사가 요구하는 논항에 특정 명사가 오거나, 부사나 부사구와 함께 사용되거나, 혹은 시제나 존칭 등을 나타내는 것을 의미한다.

목수가 의자를 나무로 만들었다. ①
목수가 의자를 망치로 만들었다. ②

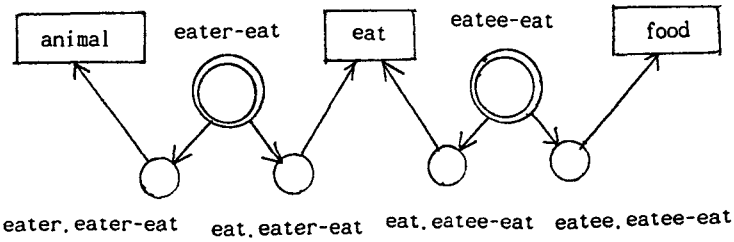
단어의 의미를 요구하는 논항의 격이라고 가정하자. 특수한 경우를- ①에서 "나무"가 '도구'의 의미가 되고, ②에서 "망치"가 '재료'의 의미가 되는 경우- 제외한다면 ①, ②에서 사용되는 "만들다"의 구체적인 의미는 조금 다르다. 동사 "만들다"가 논항으로 명사+로 를 취할 때, 이것의 의미가 어떤 경우에는 '재료'가 되고, 어떤 경우에는 '도구'가 되는지 어떻게 알 수 있을까? 이와 같은 정보는 구문 구조 분석이나 의미 분석에서 반드시 필요하다. 이에 대한 정보를 구축하기 위해 VAR는 동사가 사용되는 환경이 조금이라도 다르면 다른 Relation으로 활성된다고 가정한다. 그 후에, 언어학적인 면이나 혹은 실용적인 면에서 서로 같은 구체적인 의미를 가지게 된다고 인정되면 그 환경을 정보로 구축한다. 만약 서로 다르다고 생각되면 왜 그런지 그 이유를

고찰해서 정보를 구축한다.

VAR는 지식 베이스를 만들기 위한 기초로 고안되었으며 지식 표현 언어에서 많은 정보를 얻었다. VAR의 특성을 비교하기 위해 다음 절에서 지식 표현 언어를 간단하게 소개 한다.

2.2 KODIAK (Keystone to Overall Design for Integration and Application)

KODIAK은 absolute, relation, aspectual 의 세 가지 노드를 가진다. 이 노드들은 링크로 연결되며, 링크에는 8 가지 종류가 있다. absolute는 어떤 개체를 나타낸다. "책", "기차", "사람", "꿈" 등의 객체와 "가다", "먹다" 등의 동작 개념 등도 KODIAK에서 absolute로 나타난다. 각 relation은 항상 두 개의 aspectual을 가지며 aspectual은 relation의 형식 매개변수(formal parameter)역할을 한다. relation은 이중 원으로, aspectual은 단일 원으로, absolute는 직사각형으로 표시한다.

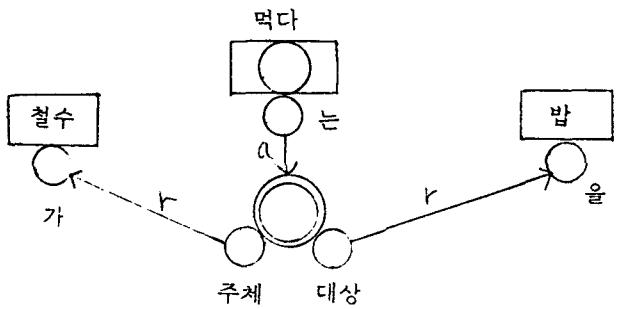


위 그림에서 eater-eat relation은 "animal"과 "eat" absolute의 관계를 나타내며, eater.eater-eat, eat.eater-eat aspectual로 각 노드들을 연결한다.

2.3 VAR 설계

VAR는 말뭉치를 구조화하기 위해 사용되기 때문에 말뭉치에 있는 모든 문장을 나타낼 수 있어야 한다. 또 VAR로 표현된 문장에서 보통 문장을 만들 수 있어야 한다. 그렇기 때문에 VAR는 문장 구조에 대한 정보도 의미 정보와 함께 효과적으로 나타낼 수 있어야 한다.

2.3.1 기본 지식



「철수가 밥을 먹는다」

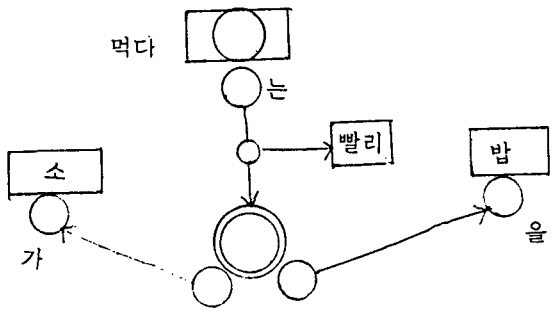
위의 그림에서 동사 "먹다"는 [◯] 으로 표시된다. [◯]에 붙어 있는 ◯ 는 동사가 활성되는 조건을 나타낸다. 여기에는 시제, 존칭 등을 나타내는 선어말 어미와 어말 어미에 대한 정보가 들어 있다. 동사로부터 활성되는 Relation노드는 동사 노드와 a(ctivate) 꼬리표가 있는 링크로 연결된며 ◎ 와 같이 표시한다. Relation 노드에 붙어 있는 ◯ 는 Relation이 요구하는 매개 변수(parameter)가 있어 이 Relation의 환경을 나타낸다. 위 그림에서 Relation노드는 '주체'와 '대상'을 매개 변수로 요구한다. 이 매개 변수는 r(equire) 꼬리표 링크로 명사 노드와 연결된다. [ ] 는 명사 노드를 표시한다. [ ] 에 있는 ◯ 는 조사가 있어 Relation에 참가하는 지위를 부여한다. Relation이 요구할 수 있는 매개 변수의 종류는 조사가 지니고 있는 의미의 종류와 같다.

2.3.2 KODIAK과 비교

VAR는 문장 표현 언어이고 앞절에서 소개한 KODIAK은 지식 표현 언어라는 점에서 이 둘은 근본적인 차이가 있다. 그러나, 대개의 지식은 문장으로 표현될 수 있다고 가정한다면 간단하게 비교할 수 있다. 우선, 두 언어 모두 Relation 중심의 망 구조를 갖는다. KODIAK의 Relation은 명사와 동사 사이에 존재하고, 특별히 만들어 주어야 한다. 그러나, VAR의 Relation은 명사와 명사 사이에 위치하며 동사에서 활성된다. 이는 VAR의 가장 중요한 특성이며 VAR의 모든 성질은 여기에서 나온다. KODIAK은 추론 시스템의 지식 기반이기 때문에 추론을 위한 다양한 링크가 있다. VAR는 문장의 정보를 표현하고 다시 문장을 생성하는 것이 주요 목표이기 때문에 조사, 어미 등이나 구문 구조를 나타낼 수 있는 방법이 있다는 점에서 많은 차이가 있다. VAR가 지식 표현 언어로써 발전하기 위해서는 다양한 링크가 도입되어야 한다.

2.3.3 성분 부사

성분 부사는 동사를 수식하기 때문에 VAR에서 부사는 동사가 Relation으로 활성되는 과정에 참여하도록 한다.

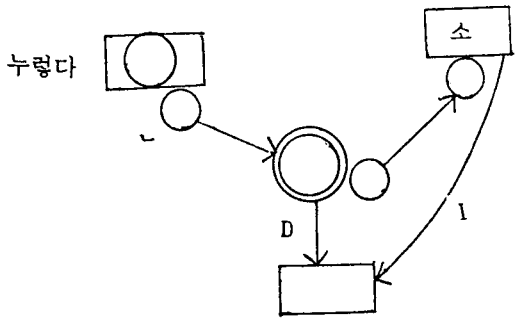


「소가 쿨을 빨리 먹는다」

위 그림에서 "먹다"가 Relation으로 활성될 때 "빨리"라는 부사가 참가할 수 있도록 만든다. 이제 부사는 동사가 활성하는데 있어 제약 조건(Restriction)으로 작용한다. 모든 활성 링크는 수의 이 있으며 아무런 조건이 없을 때에 NULL Restriction을 받는다

고 하며 이때는 대개 생략해서 간단하게 나타낸다.

2.3.4 형용사 수식을 받는 명사

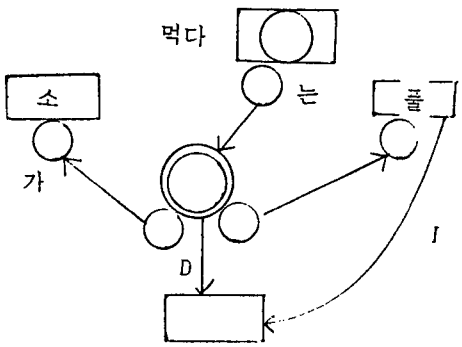


「누런 소가 ~」

위 문장의 "소"는 일반적인 "소"보다 구체적인 개념을 나타낸다. 다시 말하면, 소가 가질 수 있는 일반적인 특성 중에서 색깔만큼은 누렇다는 것을 의미한다. 위의 그림에서 볼 수 있듯이 새로운 링크가 소개된다. D-링크는 Relation에서 명사 노드로 유추(Derive)되는 것을 나타낸다. 새로 유추된 노드는 특성을 전달받을 노드에서 I-링크로 연결된다. '소가 누렇다'라는 Relation에서 새로운 노드가 유추되는데 그 노드는 소의 성질을 전달(inherit) 받는다. 이 유추된 노드의 의미는 소는 소인데 누런 소라는 것이다.

2.3.5 관형절

형용사와 비슷하지만 명사의 의미를 구체화 시키는 것에는 관형절이 있다.

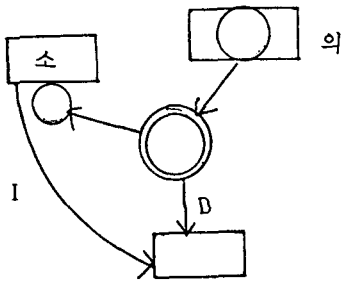


「소가 먹는 풀이 ~」

위 그림에서 만약 I-링크가 "소" 노드로부터 온다면 위 그림의 의미는 '풀을 먹는 소'가 된다.

2.3.6 조사 '의'

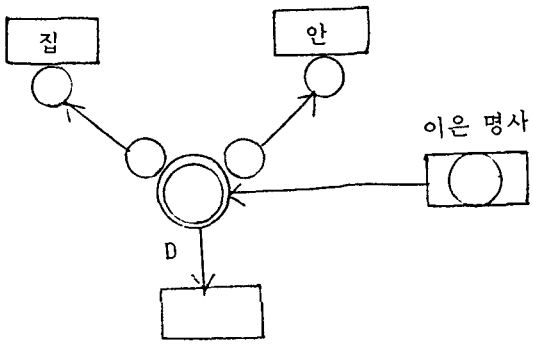
조사 '의'는 두 명사를 이어주는 조사이다. 앞에서 살펴본 형용사나 관형절과 같이 뒤의 명사를 구체화하는 역활을 한다. 그래서 '의'는 동사가 아니지만 동사와 비슷하게 Relation으로 활성된다.



「소의 조상은 ~ 」

2.3.7 이은 명사

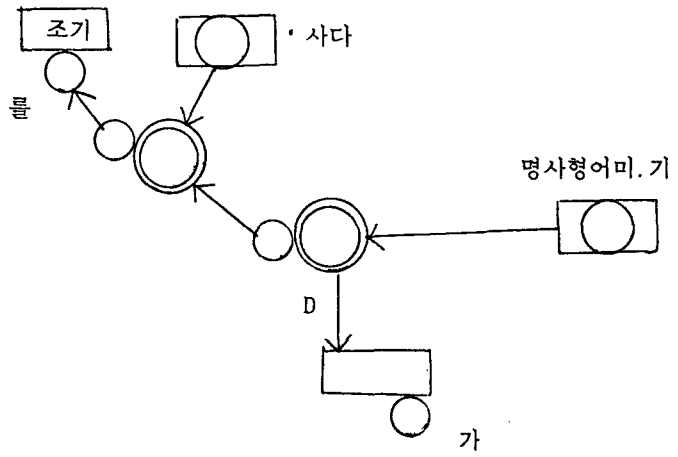
이은 명사는 두 명사가 조사의 도움 없이 연결되어서 나오는 것을 말한다. "연세대학교", "집 안"등이 이은 명사가 된다. 이은 명사는 두 개 또는 그 이상의 명사가 보이지 않는 어떤 기능어가 활성한 Relation에 의해 연결된다고 가정한다.



「집 안」

2.3.8 명사형 어미

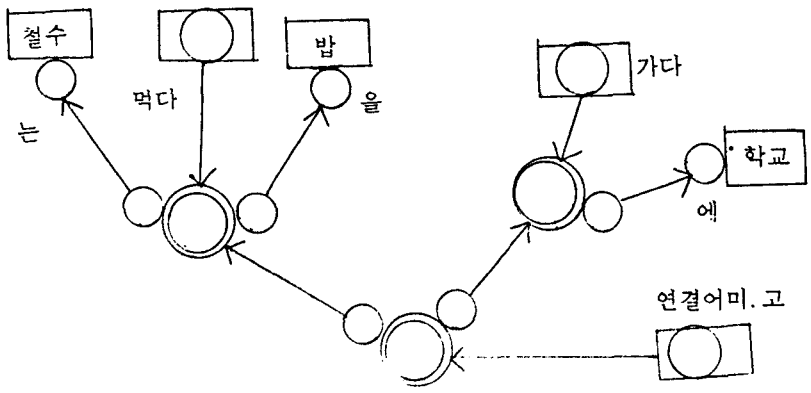
명사형 어미는 동사 뒤에 붙어서 동사가 문장 내에서 명사와 같은 역할을 하게 해 주는 것이다. 명사형 어미도 동사와 비슷하게 활성해서 Relation 노드가 되지만, 동사와는 다르게 매개 변수로 명사 노드를 받지 않고 Relation을 받는다.



「조기를 사기가 ~」

2.3.9 연결 어미

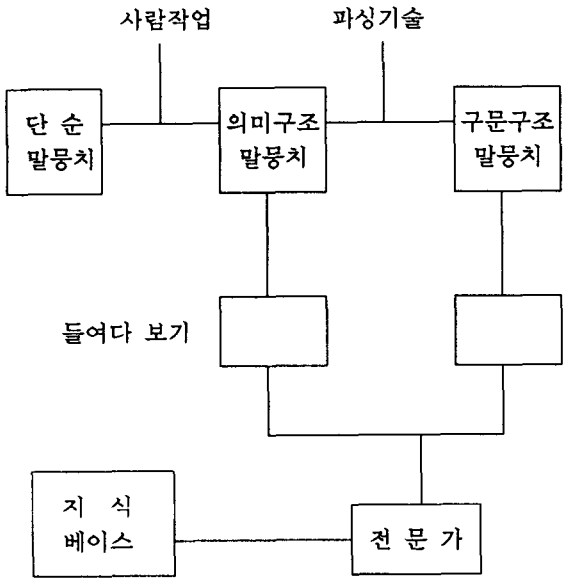
연결 어미는 두 문장을 이어준다. 명사형 어미와 비슷하게 연결 어미는 두 Relation을 매개 변수로 받는 Relation으로 활성하는 특별한 노드로 정의된다.



「철수는 밥을 먹고 학교에 갔다」

3. 의미 구조 말뭉치 전체 시스템

이제까지 VAR가 어떻게 개발되었으며, 어떻게 문장을 표현하는지를 알아 보았다. 이 장에서는 VAR로 표현된 말뭉치를 포함하는 전체 시스템에 대해 알아 본다. 의미 구조 말뭉치는 단순 말뭉치로부터 만들어진다. 이 과정에서 사람의 작업이 많이 요구된다.



3.1 입력기

의미 구조 말뭉치는 다양한 정보를 가지고 있어 이용의 가치가 높은 반면 구축하기가 매우 힘들다. 단순 말뭉치는 문자열을 입력하기만 하면 되지만 의미구조 말뭉치는 전문가가 문장을 완전히 분석한 다음 이를 입력기가 처리할 수 있는 적당한 형식으로 고쳐 주어야 한다.

```
N0 2 별 이 N2 N3
N1 4 푸르다 1 ㄴ N2 NULL
N2 0 N1 NULL 0 N3 1 0 N0
N3 3 N0 N2 N5
N4 4 반짝이다 1 ㅂ니다 N5 NULL
N5 0 N4 NULL NULL NULL 2 0 N3 1 N6
N6 2 하늘 에 N5 NULL
```

「하늘에 푸른 별이 반짝입니다」

현재 시스템이 위 문장의 VAR 표현을 입력받기 위해서는 위와 같은 입력 형식이 필요하다. 입력기는 앞으로 위의 방식과 더불어 그림 형태로 있는 VAR 표현 방식을 읽어 들일 수 있도록 발전되어야 한다.

3.2 문장 생성 알고리즘

의미구조 말뭉치는 이용자가 원하는 정보를 탐색하려 할 때, 전체 의미망을 뒤져서 알맞은 작은 의미망을 찾을 수 있어야 한다. 이용자에게는 문장형태로 복원해서 보여주어야 하기 때문에 이를 다시 문장으로 바꾸어 주는 방법이 있어야 한다.

3.3 들여다 보기

들여다 보기는 의미구조 말뭉치가 갖고 있는 정보를 효과적으로 이용자가 이용할 수 있도록 지원하는 것이다. 단순 말뭉치에서는 단어의 빈도나 단어의 용례 정도의 정보를 이용할 수 있다. 그러나, 말뭉치가 의미적으로 구조화되면 다음과 같은 다양한 정보를 이용할 수 있다. 조사 "로"가 '이동'의 의미로 사용될 때 같이 나오는 동사는 어떤 것이 있나? 동사 "가다"와 함께 조사 "로"가 사용될 때 조사 "로"가 '수단'의 의미로 사용되는 명사는 어떤 것이 있나? 들여다 보기는 전문가의 도움을 받아 범용적으로 사용될 수 있도록 발전되어야 한다.

3.4 구문구조 말뭉치

의미구조 말뭉치가 완성되고 구문 구조 분석기가 개발된다면 의미구조 말뭉치로부터 구문구조 말뭉치를 자동적으로 만들 수 있을 것이다. 구문구조 말뭉치는 의미구조 말뭉치와 다른 많은 정보를 제공할 수 있으리라 생각한다.

4. 결 론

전체 말뭉치를 모두 VAR로 나타내고 이를 관리하기 위해서는 많은 시간과 노력이 필요하다. 체계적이고 실용적인 단어의 정보를 구축하기 위해서는 그 정도의 노력은 필요하다고 생각된다. 말뭉치의 크기가 커짐에 따라 시스템의 전체 구조는 매우 달라지기 때문에 아직까지 실용적인 크기의 말뭉치를 관리할 수 있는 관리자를 개발하지는 못했다.

그러나, 작은 말뭉치를 대상으로 전체 시스템을 실험한 결과 VAR 언어는 문장이 지니고 있는 정보를 효과적으로 표현할 수 있었다. 또 간단한 알고리즘으로 VAR에서 문장을 다시 만들 수 있기 때문에 문장 표현 언어로 적당하다. VAR로 표현된 의미 구조 말뭉치는 전산학자, 언어학자에게 다양한 정보를 제공해 지식 베이스를 만드는 데 효과적인 기초가 될 수 있다.

5. 참고 문헌

[1] 이기동, 김종도, 인지 문법, 한신 문화사, 1991

[2] 이상섭, 남기심 외, 사전 편찬학 연구 1 집 - 4 집, 탑 출판사

[3] 이익환, 의미론 개론, 한신 문화사, 1989

[4] Brachman, R.J. and Schmolze, J.G., "An overvier of KL-ONE Knowledge Representation System", Cognitive Science 9(2), 1985, 171-216

[5] Dongyul Ra, ON INTERLEAVING SYSTAX AND SEMANTICS IN PARSING, Ph.D. thesis, Department of Computer Science, Michigan State University, 1989, 82 - 102

[6] Jackendoff, R.S., Semantic Structures, The MIT Press
Cambridge, Massachusetts London, England, 1990, 1 - 41
[7] Moens, M. and Steelman, M. "Temporal Ontology and Temporal Reference",
Computational Linguistic 14(2), 1988
[8] Norvig, P., Unified theory of inference for text understanding,
Ph.D. thesis, Department of Computer Science, University of California,
Berkeley, 1986, 53 - 81
[9] Wilensky, R., "Some Problems and Proposals for Knowledge Representation",
Report No. UCB/CSD 87/351, Computer Science Division,
University of California, Berkeley, 1987
[10] Woods, W.A., "What's Important About Knowledge Representation",
Computer 16(10), 22-29