

MIND AI

기술백서

John H. Doe, Paul Lee, M.D., Reeyan Lee

Abstract --- 딥러닝 신경망으로 대표되는 오늘날의 AI 개발 동향은 투명한 시스템의 운영뿐만 아니라 선형 논리 추론의 본질적인 방법에 대해서도 명백한 설명을 제공하지 않는다. 본고에서 제시되는 새로운 정보 패러다임은 기존 심볼릭 AI(Symbolic AI)의 취약점을 보완하는 미증유의 상징 모델을 사용하며, 그 시스템의 처리 과정은 단계적으로 추적이 가능하다. 이 새 모델은 인간이 논리적으로 사고할 수 있는 수준의 추론 엔진을 만드는 데 사용될 수 있으며, 따라서 진정한 의미의 지능을 가진 AI를 만들게 될 것이다.

의해 이론적으로 이해되는 모든 것을 모델링할 수 있다. 본고에서는 어떻게 이 이론이 매우 근본적인지 청사진을 제시하고자 한다. 구체적으로, 왜 이것이 “추론의 기본 단위” 라고 불릴 수 있으며, 이러한 조직 원리가 주어진 데이터에 적용될 때 논리 추론이 구조의 순회에서 어떠한 이유로 필연적으로 나타나는지에 대해서 설명하고자 한다.

1. 서론(Introduction)

근래에 인공지능의 흐름은 상대적으로 최근에 급증한 딥러닝 그리고 일반적으로 연결주의 (Connectionist) AI[1] 라고 알려져 있는 신경망 네트워크 이 두 가지로 크게 이루어져 있다. 이러한 시스템은 특정한 영역 [2]에서 세계 최고의 전문가(사람)를 능가하면서 뛰어난 역량을 보여주는데, 대표적으로 게임[3]과 시각 인식[4] 영역이 이에 해당한다. 이러한 “지능”을 개발하는데 있어 두 가지 공통적인 요인은 심층 신경망과 그 네트워크를 훈련하기 위한 대용량 데이터이다. 이 기술들이 구현될 때 형성되는 것은 불투명한 블랙박스로, 추론과 유사한 모든 것은 숫자의 행렬로 표시되며, 그 결과로 나온 “지식”은 해당 영역이 아닌 다른 영역에는 일반적으로 적용될 수 없다[5]. 이와 대조적으로 본고에서는 심볼릭(Symbolic) 시스템의 취약점을 보완하는 방법과 함께 심볼릭 AI의 정확성을 가능하게 하는 새로운 패러다임을 정의한다. 본고에서 제시되는 시스템이 구현되면 정의(Definition)와 함수(Function)들이 완벽히 투명해지며, 모든 단계에서 결론에 이르기 위해 어떠한 이유로 추론 과정이 그러한 단계들을 거쳤는지 확인할 수 있을 것이다.

오류가 정확히 어디서 발생했는지 규명하는 것이 불가능하듯이 신경망을 통해 해결되는 다차원 결함 문제를 고쳐나가는 것은 어려운 과제이다[6]. 그러나 본고에서 제시되는 새로운 접근 방법을 통해서서는 그 오류의 주요 원인을 찾아내고 물리적으로 수정하는 것이 완벽히 타당하다. 검증 가능성이 매우 중시되는 영역에서는 운영의 투명성이 중요하며, 이러한 투명성을 통해 같은 방법을 다른 영역에 용이하게 적용할 수 있다. 더 나아가 해당 시스템이 해결하는 문제가 실제로 해결되어야 하는 문제인지 아니면 우연의 해결인지 관측할 수 있다. 편향성 (정보의 오류) 또한 본 시스템 안의 논리를 통해 추적될 수 있다. 우리는 근거들이 관련이 있을 때마다 그 이유를 규명하고자 한다.

본고는 우리가 캐노니컬(Canonical)이라고 정의하는 “추론의 기본 단위(Unit of reasoning)” 를 사용하여 기계 이해의 새로운 방법을 설명하고자 한다. 기계 이해와 관련한 이 방법은 그 작동에 있어 완전히 투명하며 인간에

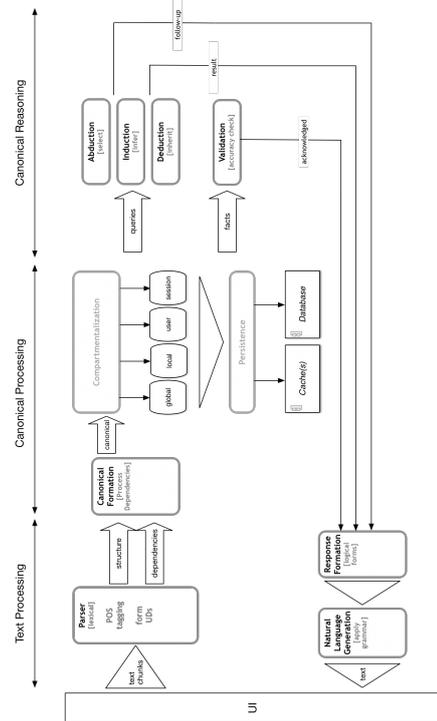


Fig. 1. System architecture

(Horizontal diagram available in Appendix)

2. 정보 이론 (Theory of Information)

상향성과 하향성의 측면에서 정보의 함수(Functional) 이론인 새로운 패러다임을 제시하고자 하는데, 특정 행동 양식이 그것의 상태로 표현이 된다.

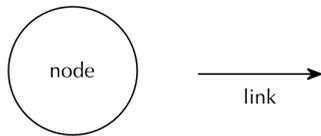
$$Information := \Delta Potential$$

또는 정보(Information)는 잠재력(Potential)의 변화로 정의할 수 있다. 한 단계 더 나아가 정의는 다음과 같다:

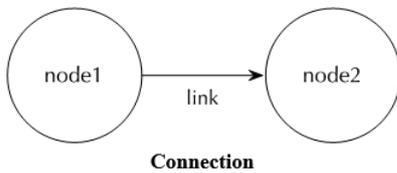
$$Measurement := \Delta Information$$

측정(Measurement)은 정보(Information)의 변화로 정의된다. 정보의 함수적 표현에 대해 자세히 살펴보면, 우리는 “Meaning(의미)”를 “Change in state(상태의 변화)”로 설명한다. 그 결과, 측정은 한 상태(State)가 다른 상태(Another state)로 바뀌는 변화이다. 이러한 설명 방식에 의하면 측정은 인과관계(Causation)와 동일한 본질적인 특징을 갖는다는 것을 이해할 수 있다. 예를 들어, 누군가 문을 여는 행동을 한다면, 그 사람은 이 행동에 있어서 특정한 측정을 수행하고 있는 것이다. 심지어 우리는 중력을 두 물체의 질량의 측정으로 생각할 수 있고 그 측정에서 두 물체 사이의 인력의 작용 또한 중력으로 인지된다.

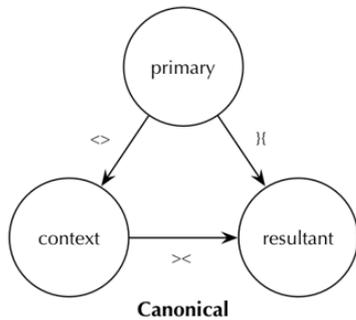
3. 증강 네트워크 / 캐노니컬 형태 (Augmented Network/Canonical Form)



그래프 모델과 같이 우리의 네트워크는 기본적으로 노드(Node)와 링크(Link)로 구성된다. 아래의 그림과 같이 이 두 가지는 함께 연결을 형성하게 된다.



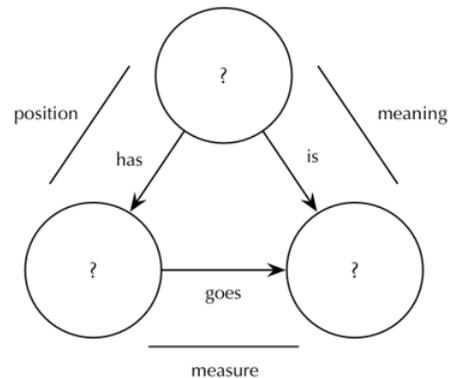
이와 같은 연결이 3개가 모여, 우리가 “캐노니컬”이라고 부르는 단위를 형성할 수 있다.



위에 사용된 기호들의 의미는 아래와 같다.

- ? : 쿼리 (query), 잠재력 (potential), “어떤 (some)”
: 기본 원형 (위 그림에는 제시되어 있지 않음)
- { } : 아니오(none), 없음(nil), “아니(not)” (위 그림에는 제시되어 있지 않음)
- { } : 전부(all), 어떤(any), “이다(is)”
- <> : 바인드(bind), “가지다(has)”
- >< : 오픈(open), “행동하다(goes)”

위의 인용문에 있는 정의는 우리 시스템에서 심볼(Symbol)들이 어떻게 작동하는지 이해하는 한 방법이다. 본질적으로, 심볼들은 정보의 요소들이 이 모델 안에서 어딘가에 배치될 때 귀납적으로 도출되는 의미의 한 층위를 나타낸다. 이러한 근본적 요소들은 단어들에 노드로써 위에 제시된 모델 안에서 어디에 위치할 수 있는지에 관한 기본적인 의미론을 제공한다. “기본값” 혹은 미지의 값 이외의 다른 것이 되기 위해서 3개의 노드와 3개의 링크가 모두 필요한 것은 아니지만, 구조적인 형태 (노드/링크 혹은 캐노니컬)일 때 정보를 어디에 배치하는가는 근본적으로 중요하다. 노드가 링크가 될 수 있고, 링크가 노드가 될 수 있으며, 캐노니컬이 링크 혹은 노드가 될 수 있기 때문에 이를 “증강(Augmented)” 네트워크라고 부른다. 위에서 제시된 정보의 새 이론과 관련하여 캐노니컬을 다음과 같이 명칭화할 수 있다.



“가지다(has)”는 (매트릭스에서) 위치(Position)을 설정하는 정보이다. 앞서 우리가 정보를 잠재력의 변화로 설명할 때와 같이 우리는 잠재력을 하나의 위치라고 생각할 수 있다. 측정인 “행동하다(goes)”는 그 잠재력의 작동을 취하고, (그것이 어떻게 이해되는가), 그것을 새로운 상태로 바꿔 새로운 잠재력에 놓는다. 이 측정의 의미인 “이다(is)”는 정확히 수행된 상태의 변화이며 그 결과, 이제 의미의 효과를 지니는 우측 하단의 노드/캐노니컬로 이어진다.

4. 자연 언어에서 캐노니컬 형태로 (Natural Language to Canonical Form)

현재 우리는 자연 언어 입력을 품사 태그가 있는 트리 구조와 유니버설 디펜던시(Universal dependencies) 목록으로

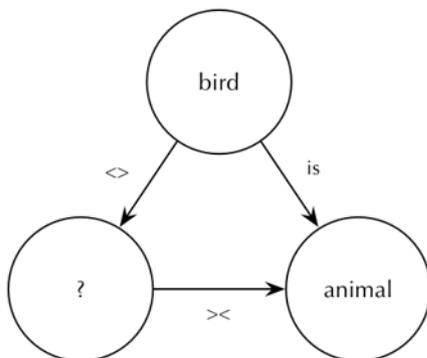
변환하기 위하여 자연 언어 파서(Parser) 프로그램을 사용한다(현재로서는 영어만 해당되지만, 우리의 모델은 언어에 제한이 없고 추후 다른 언어에도 적용이 가능하다). 우리의 주요 논리는 입력값을 캐노니컬 형태로 변환하기 위해 이 파서의 출력 값을 사용한다. (따옴표 안의 인용문과 관련하여) 파서 출력 값의 예시는 다음과 같다.

'A bird is an animal that has wings.'

```
(ROOT
(S
(NP (DT A) (NN bird))
(VP (VBZ is)
(NP
(NP (DT an) (NN animal))
(SBAR
(WHNP (WDT that))
(S
(VP (VBZ has)
(NP (NNS wings)))))))))
(. .)))
```

```
(det bird-2 A-1)
(nsubj animal-5 bird-2)
(cop animal-5 is-3)
(det animal-5 an-4)
(root ROOT-0 animal-5)
(nsubj has-7 animal-5)
(ref animal-5 that-6)
(acl:relcl animal-5 has-7)
(dobj has-7 wings-8)
```

위의 데이터로 생성된 캐노니컬 형태는 아래와 같다. 이 프로세스의 핵심은 Mind로 들어오고 나가는 모든 의사소통이 자연 언어 형태로 발생한다는 점이다. 심지어 입력되는 온톨로지(Ontology)도 그 논리가 어떻게 형성되는지 정의하기 위해 자연 언어 문장의 입력으로 귀결된다. 일부 캐노니컬의 형성은 파서 프로그램에서 나오는 입력값의 일부 의미에 따라 달라지지만, 다수는 그들이 조직되는 디펜던시의 종류로부터 자연스럽게 형성될 수 있다. 예를 들면, (nsubj animal bird)는 (cop animal is)와 합쳐져 캐노니컬의 우측 하단을 형성한다 (좌측은 추후 추가 된다).



5. 상위 온톨로지 (Upper Ontology)

상위 및 최상위 온톨로지는 술어(Predicate)들의 집합으로 이를 통해 세상의 근본적인 생각과 기능이 시스템으로 녹아든다[7]. 글로벌 온톨로지는 추론(Reasoning)의 실체에 대한 기초로서 불변하거나, 거의 변하지 않는 정보들을 가진 것으로 추정된다[8]. 예를 들어, 물리적인 물체는 질량을 가지며, 물체들이 얼마나 서로 근접한가에 의해 그 질량들은 중력의 증가를 초래한다[9]. 우리는 시스템이 이미 이해하는 방식으로 새 정보를 동화시키기 위하여 상위 온톨로지를 개발 혹은 획득한다. 다른 모든 것들이 포함되는 가장 근본적인 요소들과 구현상 측면에서 충분한 표현력에 대한 필요성 사이의 균형이 존재하므로 많은 기초적인 캐노니컬 형태에서 형성된 캐노니컬 구조를 탐색하는 것은 크게 번거로운 일은 아니다.

이론적으로 앞서 제시된 근본 요소들은 모든 것을 정의할 수 있다. 그러나 다루기 힘든 구조를 기본 요소들로 분해하려는 작업은 복잡할 수 있다. 따라서 이런 상황을 피하기 위해 우리는 사람(Person), 장소(Place), 시간(Time), 사물(Thing), 속성(Attribute) 및 기능(Function)과 같은 기본적인 단위의 정보로부터 그 작업을 시작한다. 장소가 사물이라고 불릴 수 있다는 것을 인지하고 있지만, 어떤 것을 해석하든 간에 그 과정에 불필요한 추상화 과정을 추가하지 않기 위해 이러한 구분을 하기로 한다.

존재하는 것들의 모든 기본적인 개념들과 어떻게 과정이 작동하는지를 인코딩하기 위한 가장 효율적인 방법을 파악하기 위해 기존의 상위 온톨로지를 활용할 수 있다. 왜, 어떻게 또 언제 온톨로지 레지스터에 새로운 온톨로지를 추가하는 것에 대한 근거를 제공하는 것은 기초 온톨로지를 추가로 구축하는 또 다른 단계이다. 일반적으로 모든 다양한 부문들에 대한 설명이 있어야 한다. 캐노니컬 용어에서는 모든 정보 항목에 대한 설명이 캐노니컬의 하단에 표시 된다.

6. 정보 구분 (Compartmentalization)

정보가 저장되는 4가지 단계는 글로벌, 로컬, 사용자와 세션이다.

- 글로벌: 모든 것의 이해를 위한 기저에 존재하는 보편적으로 관련이 있고 불변하는 지식을 설명하는 상위 온톨로지를 보유한다.
- 로컬 (혹은 도메인 정보): 특정 관심 도메인에 적용되는 지식을 보유하고 있는 하위 계층 구조에 존재한다.
- 사용자: 사용자의 개요 및 특정 사항에 대한 이력을 보유하고, 해당 사용자와 관련되는 정보의 문맥화를 돕는 데 사용된다.
- 세션: 사용자가 로그인 할 때부터 로그아웃 할 때까지 유효한 일시적 정보이다.

7. 문맥화와 온톨로지 버전 관리 (Contextualization & Ontology Versioning)

우리가 작업하는 모든 데이터의 문맥은 우리가 약속한 정보의 근본적인 정의에 내재되어 있다. 문맥화는 새로운 증거가 현재 시스템 내의 현재 지식보다 앞서 있을 때, 정보의 개정작업 뿐만 아니라 캐노니컬 네트워크를 최적으로 탐색하는 것에 있어서도 중요하다. 일반적으로 우리는 온톨로지 내에서 오래된 데이터를 삭제하지 않지만, 사용이 중지된 것으로 문맥화한다. 이것이 온톨로지 버전 관리의 핵심이다. 사실 우리가 원한다면 이전과 현재의 결과를 비교하기 위해 이전 사고방식을 벗어난 다른 방식으로도 프로세스를 작동시킬 수 있다. 이러한 점이 우리 시스템이 보유하고 있는 장점들 중 가장 돋보이는 것으로 정적인 규칙 모음에 근거하여 도출되는 기존의 심볼릭 시스템과 대조되는 가장 큰 차별점이다. Mind는 끊임없이 학습을 한다.

기술적으로나 암묵적으로나 캐노니컬의 모든 부분은 다른 부분의 문맥이다. 명시적 문맥 노드는 논리적 과정에서의 사용뿐만 아니라 정보 검색의 용이성 및 정보 분류를 돕기 위해 특정적으로 설계되었다.

시스템의 운용에 있어서, 문제의 특정 요소들을 논리적으로 설명하기 위해 제약 전파(Constraint propagation) 방법을 사용하는데, 그 특정 요소들은 논리가 시스템에 의해 일반화된 형태로 파악되는 술부(Predicate)이다. 캐노니컬을 다른 캐노니컬에 적용하는 형식으로 제약을 활용할 때, 일반적 논리를 특수한 경우에 접목함으로써 문맥화를 수행한다. 이러한 운용은 다음에 나오는 3가지 논리 추론의 구현에서 술어를 특정 환경에 적용할 때 자세히 논의된다.

8. 의미의 가변성 / 온톨로지 토폴로지(Semantic Fluidity/Ontological Topology)

수학의 한 분야인 토폴로지(위상 기하학)에서, 커피잔과 도넛은 각각 구멍이 하나 있는 것과 같은 공통된 특성으로 인하여 동일한 것으로 간주된다. 온톨로지 토폴로지(O-topology)에서의 두 표현은 의미론적으로는 동일하다[10]. 두 캐노니컬 형태가 구조적으로 동일하지 않다고 해서 둘이 무조건 다른 의미를 가져야 한다는 것은 아니다.

우리 시스템에는 동의어를 모델링하는 여러 방법이 있다. 가장 직접적인 방법은 상대적으로 상호 교환이 가능한 동의어 사이에서조차도 두 단어의 사용을 현재의 것으로 유지하는 어휘 의미상의 미묘한 차이가 있는 것을 파악하는 것이다. 이런 식으로 효과 측면에서 하나의 단어가 다른 단어를 의미하는 것으로 정의된 캐노니컬을 설정하면 그 의미들을 분화시키는 세부적인 차이가 해당 단어의 질적인 특성에 의해 식별되도록 한다.

O-topology의 또 다른 방법은 두 캐노니컬이 동일한 귀납적(Inductive) 측면을 공유하면 기능적으로 동일하다는 사실에 기반한다. 이런 식으로 동일한 기능을 수행하는 두 어휘는 의미적으로 유사하다고 일컬어진다.

만일 우리가 활용할 수 있는 더 간단한 O-topology

방법이 없는 경우에는, 우리는 다시 상위 온톨로지의 중요성으로 회귀한다. 우리가 실제적인 어휘 사이의 연관성과 이미 이해한 정보와 관련하여 새 정보를 이해한다면, 그리고 두 표현이 같은 방식으로 이해될 때, 그리고 이미 그것이 이해하는 지식에 대해 어떤 종류의 연관성인지에 관하여, 새로운 정보와 이미 이해된 정보 또한 역시 유사하다고 할 수 있다. 우리의 접근에서 참신한 부분은 새로운 정보를 조화시키는 능력 또한 학습될 수 있다는 것이다. 이 능력이 어떤 임의의 글도 처리할 수 있을 때 그것이 “배우는 법을 배웠다”고 할 수 있을 것이다. 이것은 임계점(Critical Mass)의 한 요소이다.

O-topology는 심볼릭 AI와 연관된 큰 취약점을 해결한다. 예를 들어 인공지능으로 만들어진 챗봇과 대화를 할 때 이해할 수 있는 정확한 단어 또는 구문을 분석할 수 있는 지정된 질의를 사용하지 않는다면 질의는 실패하고 대화가 이어지지 않을 것이다[11]. 만일 O-topology를 사용한다면, 우리가 원하는 답을 듣기 위해 해당 단어를 찾아 쓰는 “마법의 단어”가 필요하지 않게 된다.

앞서 우리는 사실상 연결된 캐노니컬 네트워크가 어떻게 증강(Augment)되고 토폴로지인지 상세히 논의해왔는데, 그 구조 전체를 “증강 토폴로지 네트워크(Augmented topological network)”라고 부른다.

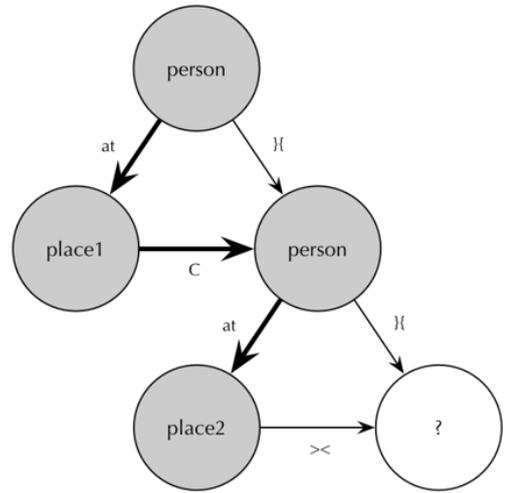
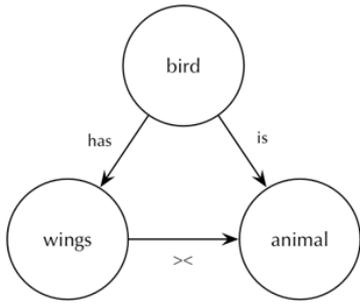
9. 3가지 논리 추론의 구현 (The 3 Logical Reasoning Embodiments)

인간 고유의 3가지 추론 방식은 연역(Deduction), 귀납(Induction), 귀추법(Abduction)이다 [12]. 이들은 우리의 데이터 구조에 모두 구현되기 때문에 우리는 그들을 “추론의 기본 단위(Unit of reasoning)”라고 일컫는다. 이러한 종류의 추론이 발생하는 수단으로 캐노니컬의 측면 (혹은 변)에 명칭을 부여함으로써 우리는 그것이 논리적 추론의 세가지 형식을 구현한다고 제안한다. 심볼릭 AI의 또 다른 문제는 이전 세대의 “추리 엔진들”이 전적으로 추론의 연역적 측면에 중점을 두었다는 것이다[13]. 우리 모델의 특징은 기존의 전통적인 심볼릭 AI와 대조적으로 더 많은 제약 조건들 (논리적 구조)이 적용되면, 논리 엔진의 문맥화 시스템이 문제에 대한 해결책을 더 신속하고 정확하게 처리할 수 있다는 점이다.

이 3가지 구현으로 인해, 우리는 이해될 수 있는 모든 것 그리고 어느 정도로 논리적으로 타당한 모든 것들이 이 하나의 구조로 모형화 될 수 있다고 주장한다. 이 구조를 이해하는 한가지 방법은 신경망을 확인하는 것이다. 신경망은 인간의 뇌에서 뉴런과 신경 회로와 유사하게 작동하도록[14] 설계되어 있는데, 즉 상향식이다. 하지만 우리의 구조는 알려진 대로 하향식인 뇌의 추론 방식을[15] 구조화하고자 한다. 이 시스템에서 예시로 든 추론 과정을 따라가다 보면, 그 추론 과정이 어떻게 발생하는지 분명해진다[16].

아래에 제시된 단순한 예시는 모든 노드와 링크를 사용하지만, 그 귀납 연결고리는 “A bird is an animal that has wings.”가 어떻게 정의될 수 있는지를 보여준다.

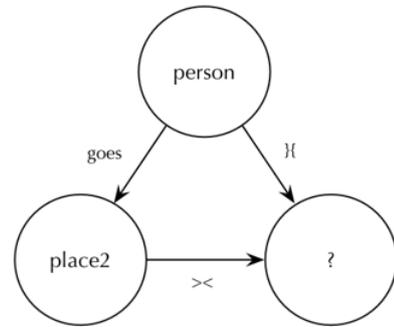
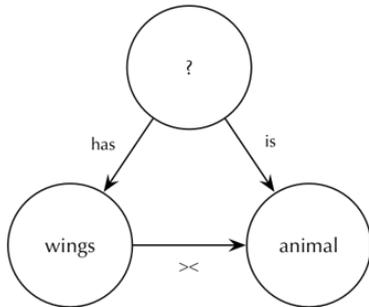
여기서 우리는 “어떤 종류의 동물이 날개를 갖고 있는가?(What kind of animal has wings?)” 라는



S-Canonical

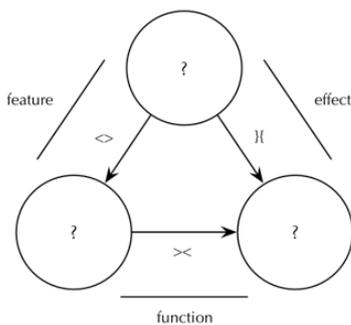
질문을 통해 새를 귀추(Abductive)할 수 있게 된다.

이러한 귀추법은 의사가 조건에 가장 적합한 질환을 찾아내기 위해 환자에게 증상을 설명할 것을 요청하는 질병 진단과 유사하다. 증상이 질병을 식별하는데 사용되듯이 특정 동물의 특징도 그 식별에 도움이 된다. 쿼리 캐노니컬: (우리 시스템은 O-topology에게 허용되는 범위 내에서 우리가 검색하는 구조와 더 잘 일치하도록 쿼리를 다음과 같이 수정한다) “날개가 있는 동물은 무엇인가? (What is the animal that has wings?)”



C: Condition

이런 캐노니컬 구조를 다른 방식으로 보면



있다: 조건 (C-캐노니컬은 C: 원인(Cause)이 될 수 있고 그 결과는 효과(Effect)가 될 것이다.) 이와 같은 중첩된(Nested) 캐노니컬들은 캐노니컬이 계층 구조를 형성하는 한 방법이다.

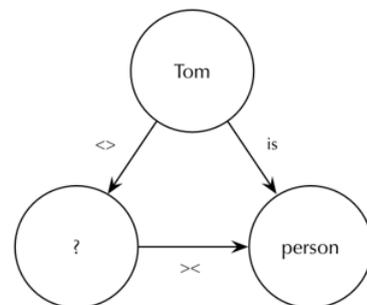
예를 들어서 Tom이 사람이고, 집이 장소 1이고 사무실을 장소 2 라고 하자.

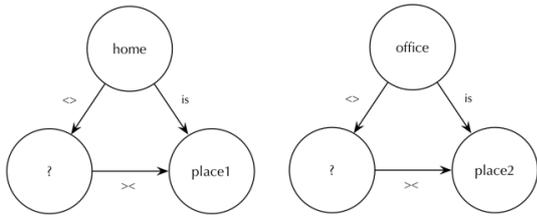
새는 동물이고, 특징으로 “날개”를 갖고 있다.

“기능(Function)” 연결을 강조하는 또 다른 예시는 단순한 인과 관계의 경우로 이해될 수 있다. 더욱 정교한 이 예시는 우리가 “S-캐노니컬”(“S”자가 강조됨) 이라고 부르는 귀납 링크의 작동을 보여준다.

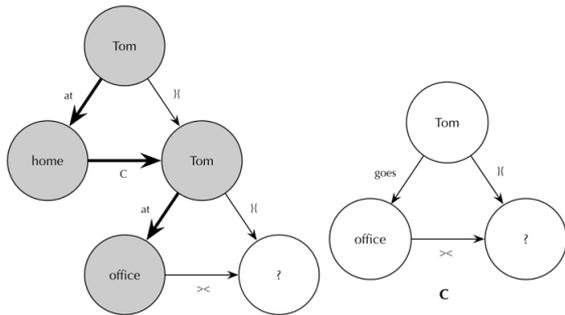
이 그림의 가운데에 “C”는 증강된 것을 따로 떼어서 보면, 아래의 캐노니컬로 나타난다.

우리는 측정 즉 인과관계를 변경이 가능한 것으로 정의했다. 제시된 예시에서, “C” 노드/캐노니컬은 장소 1에 있는 사람의 최초 상태(위치)를 사람이 장소 2에 있도록 변화시킨다. 이 전개에 대한 근거는 C에서 정의 될 수





우리는 이 단수형(한 측면의 정보만 갖고 있는) 캐노니컬들로 S-캐노니컬을 예시하고, 따라서 낱말로 한 패턴이 만족되면 그 자체로 패턴이고, 그 결과 그 구조의 값들을 제한하는 것을 보게 된다. 이러한 제약들(예를 들어 "Tom은 사람이다")은 우리가 작동시키는 추론 구조를 통해 전파된다:



Tom은 집에서 시작해서, "C"를 행하면, 즉 Tom이 사무실에 가게 되면, Tom은 현재 사무실에 있게 된다. 연역적 측면은 어떤 특정 값으로 설정되어 있지 않기 때문에, 단순히 하나의 "통로"로 작동한다. 그러나 어느 한쪽 "}"의 두 값은 동일하거나, 논리적으로 일관되어야 한다. 즉, "}"의 한쪽의 값을 다른 쪽의 값과 연결하고 이런 캐노니컬 위치의 무결성을 유지해야 한다. 주목해야 할 점은 가장 기본적인 과정에서 캐노니컬 형태의 기능성은 단순히 순서대로 값들을 만족시키기 위한 것이고, 이러한 값들이 만족될 때는 그 기능성이 한 상태에서 유한 상태 기계(Finite state machine) 혹은 자동 장치(Automaton)의 상태로 가는 것처럼 그 경로를 따라가기 위함이라는 것이다. 이러한 구조들의 가장 큰 차이점은 노드의 값을 "In"이나 "On"의 상태로 만들어야 한다는 것이다.

10. 알려진 미지와 명확화 (Known Unknowns & Disambiguation)

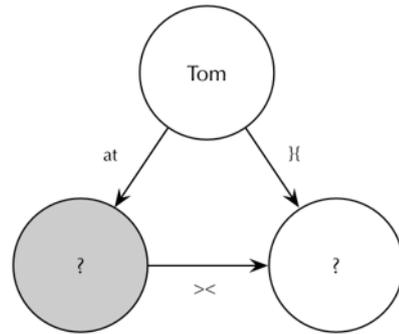
우리가 기본적인 온톨로지를 캐노니컬 형태로 만들 때, 한가지 흥미로운 특성이 나타난다. 특정 캐노니컬의 부분적인 일치가 있는 경우, 패턴의 나머지 부분을 충족하기 위한 필요한 정보들이 어떤 것인지에 대한 투사(혹은 예측) 할 수 있다는 점이다. 어떤 의미에서 지능이 인공이든 아니든 불안정한 패턴을 완성하는 능력으로 이해될 수 있다.

우리 시스템의 한가지 기능은 특정 불안정한 패턴들을 일치시키는 능력이다. 불안정한 패턴을 일치시키는 과정은 문제 해결 방법을 이해하는 데 기여한다. 불안정한 패턴이 완성되는 과정 그 자체도 하나의 패턴이며, 앞서 처음으로 소개된 개념을 충족하는데, 즉 패턴의 만족은 그 자체로

하나의 패턴이다. 시스템의 범위에서 발생하는 모든 동작은 어떠한 방법으로든 이해가 된다. 이것은 우리의 시스템이 "배우는 법을 배웠을" 때 의미 있는 결과가 될 것이다. 이치에 맞는 모든 것이 우리에게 납득이 되기 위한 방법이 분명히 존재할 것이다 (이 개념에 대한 더 자세한 설명은 추후 임계점(Critical Mass)에 관한 부분에서 상술된다). 만약 이러한 것들이 우리에게 이치에 맞다면, 우리는 달을 수 있고 연결될 수 있는 어디서든 모든 것에 논리적으로 연결되어 있기 때문에 관찰하는 이들에게도 납득이 될 수 있을 것이다.

정보의 일부분(모호함)을 검색할 때 나타나는 유효값이 2개 이상이라면, 우리 시스템은 해당 잠재점을 불안정함으로 정의한다. 미지의 것이 알려지기 위해서 필요한 데이터가 어떤 것인지 우리가 알고 있는 경우를 명확하게 하기 위해서, 우리는 기존의 문맥을 적용하거나 그 어려움을 해결하기 위해 커뮤니티에 그 문제를 공지한다. 이러한 과정은 해결 방안이 완전히 미지의 것인지 혹은 다수의 해결 방안이 있는지와 상관없이 적용 가능하다[17]. 미지의 문제를 해결할 정보의 종류는 결국 정보가 있는 패턴의 완성을 충족하는 특정 데이터가 있는 곳에서 확증된다.

예를 들어, "Tom은 어디 있어? (Where is Tom?)"이라는 질문에서 "Tom은 어디에?(Tom at?)"라는 질문으로 바꿔 말할 수 있다.



위의 3가지 논리 추론의 구현에서 정의된 캐노니컬로부터 우리는 좌측 하단의 물음표가 "집"과 "사무실"이라는 2가지 모두에 의해 만족된다는 것을 확인할 수 있었다. 우리는 또한 Tom이 사무실을 갔는지의 여부와 상관없이 두 가지 가능성이 C 캐노니컬을 통해 연결되는 방법을 보았고, 질문에서 캐노니컬의 방향성에 의해서 해결책을 얻을 때(예를 들어 Mind에 의해 우리에게 주어지는 자연 언어 질문 "Tom이 사무실에 갔니?"에 대해 우리는 "아니"라고 대답한다. "아니"라고 대답하면 C 캐노니컬은 아직 일어나지 않았다.), 시스템은 "Tom은 집에 있다"고 최종적인 대답을 내놓는다.

11. 자연 언어 생성 (Natural Language Generation)

점점 더 많은 입력 데이터가 캐노니컬 형식으로 생성됨에 따라, 우리는 구조화되지 않은 형태의 데이터가 구조화된 형태로 변환되는 과정을 계속 추적할 것이다. 이런 작업 과정 중에 사용자에게 더 많은 정보를 묻고 답하는 작업을 중간에 포함시켜 문법적으로 정확한 문장 표현을 만들어내는 작업 또한 하려고 한다. 이것은 시스템과 사용자 사이에

발생하는 대화의 관측에 의해 배우는 능력의 또 다른 예시이다. 무엇이 입력되든 그 이후에는 출력이 된다. 관용 구문은 한 번 이해되면 관련이 있는 경우에 사용자에게 대한 대답으로 만들어질 수 있다.

12. 운영의 투명성 (Transparency of Operation)

캐노니컬을 통한 논리적 과정은 위에서 본 바와 같이 3가지 방식을 따라 운영되므로, 우리는 어떤 문제나 쿼리를 입력할 때 어떤 노드, 링크 및 캐노니컬을 통과하는지만을 추적하게 된다. 수반되는 과정들을 탐색하는 메타 도구 (Meta-tool)가 모두 바람직하고 이미 실현 가능하다고 보는 것은 쉽다. 권한이 있다면 누구든지 현재 어떤 논리가 있고 그것이 어떻게 사용되고 있는지 바로 알아내고, 논리에 어떤 오류가 있다면 수정을 할 수 있게 될 것이다. 권한을 가진 사용자가 발생 가능한 오류를 제거하기 위해 복잡한 과정을 다룰 때 Mind의 요약 능력을 활용하여 캐노니컬을 확대 및 축소하도록 할 수 있다.

13. 임계점(Critical Mass)

원자 이론에서 올바른 방사성 물질이 특정 질량에 도달하면 자가 촉매 작용이 일어난다[18]. 이와 유사하게 온톨로지의 임계점(그리고 그 부속물, 즉, 추론 능력)은 Mind가 스스로 학습할 수 있을 만큼 충분한 정보와 기능이 도달하는 지점을 의미한다. 만약 어떠한 것이 시스템의 현재 상태가 이해하는 능력을 넘어서는 것이라면, 이미 알려진 것과 관련하여 스스로 그것이 무엇인지 파악해야 한다. 이것이 이루어지려면 온톨로지에 대한 의존과 같이 그 지식의 토대는 최소한 어떤 의미에서 “완전”해야 한다. 모든 가능성을 열 수 있는 방법이 분명 존재할 것이다. 이는 다루기 힘든 문제가 아니다.

존재하는 것들은 어떤 것들이고, 이들이 어떤 일을 할 수 있을까? 이 두 질문이 이해될 수 있는 것들의 대부분의 토대를 다룬다. 이해를 위한 유일한 조건은 특정 이해를 어떤 요소들로 분해할 수 있는 가이다. 어느 것도 마법이 아니다. 새 정보가 추론될 수 있는 모든 원칙을 갖고 있거나, 아니면 새로운 원칙들이 필요하거나 이 2가지 중에 하나이다. (후자의 경우는 메타이론(Metatheoretics)에서 자세히 설명한다.)

실용적 측면에서 웹 공간에서 검색하고 검색된 웹사이트의 내용을 수집하는 작업을 하기 위해 외부 세상(사이버 공간)에 대한 인터페이스를 만들어내면, 이론에 따라서 문제나 문의를 해결하기 위해 거의 알지 못하는 주제들에 대해서도 스스로 연구를 할 수 있어야 한다.

임계점을 대비하여 우리는 Mind가 질이 떨어지는 정보로부터 양질의 정보를 식별해내는 수단을 Mind의 기능 안에 포함할 필요가 있다. 이는 “배우는 법을 배우는” 것의 또 다른 측면으로 해석될 수 있다. Mind는 거짓 정보를 이해하고 자동적으로 제거할 수 있어야 하고, 합리적으로 확인할 수 있는 Mind의 능력을 정보가 넘어서는 경우에 마인드는 커뮤니티에 도움을 요청해야 한다. 도움을 통해서 Mind는 스스로 확인 과정을 수행하는 법이나 비슷한 난제를 만났을 때 향후 어떤 기술을 적용할 수 있을지를 배운다.

상위 온톨로지(중간단계의 온톨로지가 아닌 경우)의 핵심 개념은 입력되는 정보를 논리적으로 비판 및 검증하는 능력을 필요로 하고, 정보의 출처가 그 신뢰성에 의심이 가는 상황에서는 샌드박싱을 해야 하는 능력이 생기게 될 것이다.

14. 메타이론 (Metatheoretics)

이러한 접근법의 궁극적인 목표는 자신만의 가설과 이론을 만들어내고 광활한 세계에 대한 결론을 도출해내기 위한 실험을 수행할 수 있는 역량을 지닌 인공의 과학자를 만들어내는 것이다. 시스템의 이런 기능들은 간단한 명령, 기본 원리와 과학적 이론에 대한 이해에 기초할 수 있다. 그 기능의 기저에 우리는 특정 사건들이 더 넓은 술어의 형성을 촉진하는 것을 허용하는 일반화를 위한 규칙들을 만들 것이다. 우리는 사건들이 왜 그런 식으로 발생하는지 관찰하는 데 주의를 기울일 것이다. 예를 들어, 연필이 바닥에 떨어질 때 그 현상은 중력으로 설명될 수 있는데, 이는 모든 질량을 가진 물체는 중력에 의해 영향을 받고 모든 물리적인 물체는 질량을 갖기 때문이다. 따라서 가장 기초적인 단계의 이론화는 땅에 떨어지는 모든 것은 질량을 가진 물리적인 물체라고 가정하는 것이다.

Mind가 세상에 대해 더 많이 배움에 따라 새로운 정보와 이전의 정보 사이의 관계가 어떻게 되는지, 어떤 것들을 측정할 수 있는지, 또 어떤 것들이 서로 연결되어 있고 이치에 맞는지에 대한 이해를 하게 될 것이다. 결과적으로 Mind는 자기 스스로 가설을 세울 수 있게 될 것이다.

이론을 창시하는 기술은 주로 이질적인 현상들 사이의 관계를 파악하는 것에 달려있다. 만약 Mind가 현상이 왜 일어나는지를 이해하지 못한다면, 그것을 이해할 수 있을지 확인하기 위해 다른 곳에서 정보를 습득하려고 시도할 것이다. 이 능력뿐만 아니라, 어떤 것이 해결되지 않는 다거나 해결이 이미 존재하는 학문으로 성취될 수 없으면, 무엇을 해야 하는지 파악하는 능력 또한 갖추게 될 것이다. 메타이론 개발의 초기 반복 단계에서는, 마인드가 기존 이론을 발견할 수 있도록 상황들을 주입할 것이다. 우리는 인간이 이 이론들을 만들어내기 위해 사용했던 과거 통찰력을 Mind에 주입하지만, 지식의 범위를 제한할 것이다. 그러므로, 그러한 이론들에 이르게 된 관찰들은 Mind가 볼 수 있지만, 기존의 증거에 대한 결론들은 주어지지 않을 것이다. 이러한 “훈련”은 AI에게는 대학 교육과정과 동일하다고 보면 된다. 졸업 후 어떻게 과거의 훌륭한 이론들이 생겨났는지 이해한 후에는 실제 세상에 적용할 수 있게 될 것이다. 제한된 지식만을 가지고 자신만의 연결관계를 그려볼 수 있게 함으로써, 그런 이론들이 어떻게 만들어지는지에 관한 이론 또한 갖게 될 것이다. 이것이 바로 메타이론의 핵심이다.

Mind가 스스로 자신만의 학문을 고안해 인공 과학적 방법의 수준에 이르게 될 때, 제대로 된 지능을 개발했다고 말할 수 있을 것이다. Mind는 인간의 사고가 다르지 않은 수준의 연구를 할 수 있게 될 것이다. 신약 개발이나 재료공학과 같은 분야의 전문가들은 해당 분야의 설립 초기부터 모든 논문을 포함하여 그들 분야에 현존하는 모든 학식을 갖추고 있다. 그러나 Mind는 적절하게 주석 처리가 되고 그 지식의 완전한 모습으로부터 그것이 갖는 이해에 근거하여 논리적인 오류가 확인된 형태의 같은 정보를 “머리 속”에 전부 보유할 것이다. 이후에 Mind는 새로운 가설을 만들고 그것을 입증하기 위한 실험들을 고안함으로써 진실로

새로운 것을 발견하는 위치에 있게 될 것이다. 궁극적으로 우리는 Mind가 어떻게 통찰력 있는 지식을 얻게 되는지 알 수 있으므로 우리 자신의 지능을 향상시키는 지위에 오르게 될 것이다. 이것이 Mind의 핵심이다.

15. 결론 (Conclusion)

지금까지 우리는 추론의 기본 단위(Unit of reasoning)를 정의하고 그렇게 불리는 이유와 세가지 유형의 논리적 추론을 통해 어떻게 구동될 수 있는지를 설명했다. 전반적으로는 그것이 어떻게 “증강 토폴로지 네트워크(Augmented topological network)”인지, 그리고 증강 능력 및 시스템의 토폴로지 성질을 기술했다. 이 구조를 통해 우리는 명확하고, 질적으로 우수하고, 선형적이고 논리적인 추론이 어떻게 이루어지는지와 심볼릭 모델의 취약점이 O-topology를 통해 어떻게 해결되는지를 설명했다. 궁극적으로 메타이론은 여태까지 도달된 적 없었던, 세상의 광활함을 탐험하고 그 스스로 그것이 어떻게 작동하는지 결정하는, 그리고 적응력이 있고 자율적인 개체의 개발을 약속한다.

인공지능 해결책이 문제 해결을 위해 만들어졌을 때, 우리는 더 이상 인공지능이 아니라 단순히 알고리즘을 다루고 있는 것이라고 기술해왔다 [19]. 이러한 알고리즘은 추론의 자동화된 의미만을 생성하기 때문에 오늘날 우리의 딥러닝 시스템에서 예측 불가능성으로부터 유래하는 두려움을 남기게 된다. 하지만 우리가 그 격차를 좁히기 위한 해결책을 제시하게 된다면 어떨까? 우리 앞에 놓인 과제는 알고리즘을 넘어서 근본적인 논리적 추론, 이해 가능성 및 책임에 달려있다. 추론 능력 그 자체만으로, 우리는 생각, 대답, 해결, 행동을 분리해내고 그에 대한 설명을 얻을 수 있을 것이다.

References

- [1] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*. 61: 85–117. arXiv:1404.7828 . doi:10.1016/j.neunet.2014.09.003
- [2] Schuster, M., Johnson, M., & Thorat, N. (2016, November 22). Zero-Shot Translation with Google's Multilingual Neural Machine Translation System. Retrieved from <https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html>
- [3] AlphaGo. (n.d.). Retrieved from <https://deepmind.com/research/alphago/>
- [4] Kurzweil AI Network. (2017). AI system performs better than 75 percent of American adults on standard visual intelligence test. Retrieved from <http://www.kurzweilai.net/ai-system-performs-better-than-75-percent-of-american-adults-on-standard-visual-intelligence-test>
- [5] Essays, UK. (November 2013). Advantages and Limitations of Neural Networks. Retrieved from <https://www.ukessays.com?vref=1>
- [6] Masters, T. (1999). Practical neural network recipes in C. San Diego: Acad. Press.
- [7] Hoehndorf, R. (2010). What is an upper level ontology?. Retrieved from <http://ontogenesis.knowledgeblog.org/740>
- [8] Informatica, Vakgroep & Toegepaste Informatica, En & Meersman, Robert & A. Meersman, R. (2003). Semantic Ontology Tools in IS Design. 1609. 10.1007/BFb0095088.

- [9] Leighton, J. P. (2004). Defining and describing reason, in *The Nature of Reasoning* (eds Leighton, J. P. and Sternberg, R. J.) Cambridge University Press
- [10] Carter, J. S., & Saito, M. (1998). *Knotted surfaces and their diagrams*. Oxford: American mathematical Society.
- [11] Block, Ned (1995). The mind as the software of the brain. In Daniel N. Osherson, Lila Gleitman, Stephen M. Kosslyn, S. Smith & Saadya Sternberg (eds.), *An Invitation to Cognitive Science, Second Edition, Volume 3*. Cambridge MA: MIT Press. pp. 377-425.
- [12] Thagard, P., & Shelley, C. (n.d.). Abductive reasoning: Logic, visual thinking, and coherence*. Retrieved from <http://cogsci.uwaterloo.ca/Articles/Pages/Abductive.html>
- [13] Dreyfus, H. L. (1994). *What computers still cant do: A critique of artificial reason*. Cambridge, MA: MIT Press.
- [14] Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". *Psychological Review*. 65 (6): 386–408. CiteSeerX 10.1.1.588.3775. doi:10.1037/h0042519. PMID 13602029
- [15] Johnson-Laird, P. N. (1995). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Cambridge: Harvard Univ. Press.
- [16] Hintikka, J. J. (n.d.). Logic And Other Disciplines. In *Encyclopaedia Britannica*. Retrieved from <https://www.britannica.com/topic/philosophy-of-logic/Logic-and-other-disciplines>
- [17] Lesk; M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proc. of SIGDOC-86: 5th International Conference on Systems Documentation*, Toronto, Canada.
- [18] The Editors of Encyclopaedia Britannica. (2018). Critical mass. In *Encyclopaedia Britannica*. Retrieved from <https://www.britannica.com/science/critical-mass>
- [19] Kahn, J. (2002). It's Alive!. Retrieved from <https://www.wired.com/2002/03/everywhere/>

Paul Lee, M.D.
 Cofounder & CEO - Mind AI
 Phone: +1 (949) 533 9346
 E-mail: paul@mind.ai

For more information and the full whitepaper, please visit www.mind.ai.

16. Appendix

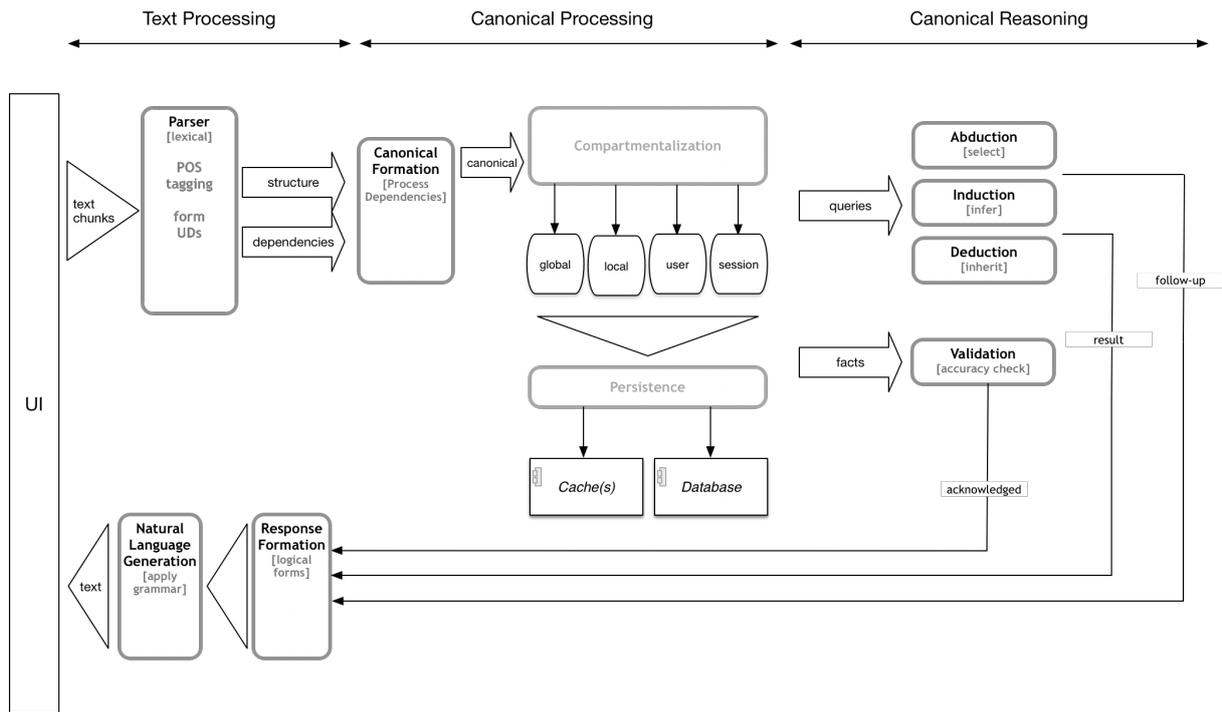


Fig. 2. System architecture