

AI-완전한 질문 답변을 향하여 : 전제 조건 장난감 작업 세트

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush,
Bart van Merriënboer, Armand Joulin & Tomas Mikolov Facebook
AI Research 770 Broadway New York, USA
{jase,abordes,spchopra,tmikolov,sashar,bartvm}@fb.com

추상적인

기계 학습 연구의 장기적인 목표 중 하나는 특히 지능형 대화 에이전트 구축과 같이 추론 및 자연 언어에 적용할 수 있는 방법을 생성하는 것입니다. 그 목표를 향한 진행 상황을 측정하기 위해 우리는 질의 응답을 통해 독해력을 평가하는 프록시 작업 세트의 유용성을 주장합니다. 우리의 작업은 시스템이 연결 사실, 간단한 귀납, 연역 등을 통해 질문에 답할 수 있는지 여부와 같은 여러 가지 방법으로 이해도를 측정합니다. 작업은 인간과 대화할 수 있는 것을 목표로 하는 모든 시스템의 전제 조건으로 설계되었습니다. 우리는 많은 기존 학습 시스템이 현재 문제를 해결할 수 없다고 생각하므로 우리의 목표는 이러한 작업을 기술 세트로 분류하여 연구자가 시스템의 결함을 식별(및 수정)할 수 있도록 하는 것입니다. 우리는 또한 최근 도입된 메모리 네트워크 모델을 확장하고 개선하여 모든 작업이 아닌 일부 작업을 해결할 수 있음을 보여줍니다.

1 소개

신경망에 동기를 부여하는 데 도움이 되는 XOR 문제(Minsky & Papert, 1969; Rumelhart et al., 1985)부터 다음 중 일부에 동기를 부여하는 데 도움이 되는 원형 및 고리 데이터 세트에 이르기까지 기계 학습에서 합성 작업을 사용한 풍부한 역사가 있습니다. 가장 잘 알려진 클러스터링 및 준감독 학습 알고리즘(Ng et al., 2002; Zhu et al., 2003), 시계열에 대한 Mackey Glass 방정식(Müller et al., 1997) 등 - 사실 잘 알려진 UCI 데이터 세트 중 일부(Bache & Lichman, 2013)도 합성 데이터입니다(예: 파형). 최근 작업은 이러한 추세를 계속합니다. 예를 들어, 메모리 구성 요소를 사용하여 학습 알고리즘을 개발하는 영역에서 합성 데이터 세트는 Graves 등의 Neural Turing Machine을 개발하는 데 사용되었습니다. (2014) 및 Weston et al.의 메모리 네트워크. (2014), 후자는 이 작업과 관련이 있습니다.

합성 데이터에 대한 관심의 이유 중 하나는 이를 사용하여 새로운 기술을 개발하는 것이 더 쉬울 수 있다는 것입니다. 많은 양의 실제 데이터("빅 데이터")로 작업하는 것은 "간단한 모델과 많은 데이터가 더 적은 데이터를 기반으로 하는 더 정교한 모델보다 우선"하므로 연구원을 더 단순한 모델로 이끄는 경향이 있다는 것은 잘 알려져 있습니다(Halevy et al., 2009).). 예를 들어 언어 모델링을 위한 N-gram은 기존의 경쟁 방식에 비해 잘 작동하지만 텍스트를 진정으로 이해하는 모델과는 거리가 멀다.

연구원으로서 우리는 알고리즘 공간에서 로컬 최소값에 갇힐 수 있습니다. 합성 데이터의 개발은 그것을 시도하고 깨는 한 가지 방법입니다.

이 작업에서 우리는 텍스트 이해 및 추론을 위한 학습 알고리즘 개발을 돕기 위한 프레임워크와 일련의 합성 작업을 제안합니다. 인공지능의 장기적 목표인 일반적 대화에서 에이전트의 성능을 자동으로 평가하는 것은 상대적으로 어렵지만 입력된 질문에 대한 응답, 즉 QA(Question Answering) 작업을 평가하는 것은 상대적으로 쉽습니다.

질문 답변은 엄청나게 광범위합니다. 생각할 수 있는 모든 작업을 이 설정에 포함할 수 있습니다. 이를 통해 공통 프레임워크에서 학습 알고리즘의 다양한 기능을 테스트하는 다양한 작업의 광범위한 세트를 제안할 수 있습니다.

우리의 작업은 고전적인 텍스트 어드벤처 게임(Montfort, 2005)과 유사한 물리적 세계의 통합된 기본 시뮬레이션으로 구성되어 있습니다.

서로 서로 함께. 시뮬레이션이 실행되면 기반 텍스트와 질문 답변 쌍이 동시에 생성됩니다. 우리의 목표는 다양한 종류의 질문을 우리의 작업이 되는 기술 세트로 분류하는 것입니다. 우리의 희망은 이러한 작업에 대한 성능 분석이 현재 모델의 약점을 드러내고 이러한 약점을 완화하는 새로운 알고리즘 설계에 동기를 부여하는 데 도움이 되는 것입니다. 우리는 이를 새로운 모델을 깨기 위해 아마도 적대적인 방식으로 응답하여 새로운 작업을 설계할 수 있는 피드백 루프로 이것을 구성합니다.

우리가 설계한 작업은 섹션 3에 자세히 설명되어 있고 이를 생성하는 데 사용된 시뮬레이션은 섹션 4에 나와 있습니다. 섹션 5에서는 작업에 대한 표준 방법의 벤치마크 결과를 제공하고 성공과 실패를 분석합니다. 우리가 생각하는 알고리즘 개발과 작업 개발 사이의 일종의 피드백 루프를 제시하기 위해 섹션 A에서 QA에서 유망한 성능을 제공하는 것으로 나타난 최신 메모리 네트워크 방법에 대한 일련의 개선 사항을 제안합니다. 우리는 우리가 제안한 접근 방식이 실제로 일부 작업에서 향상된 성능을 제공하지만 아직 해결되지 않은 문제 중 일부를 해결할 수 없음을 보여줍니다.

2 관련 작업

QA 기반 전략을 사용하여 언어 이해를 목표로 하는 여러 프로젝트가 최근 등장했습니다. 대화나 요약과 같은 작업과 달리 QA는 평가하기 쉬우므로(특히 참/거짓 또는 객관식 시나리오에서) 매력적인 연구 방법이 됩니다. 문제는 질문의 정의에 있습니다. 질문은 성인 인간(또는 어린이)이 명확하게 대답할 수 있어야 하지만 여전히 약간의 생각이 필요합니다. Allen Institute for AI의 주력 프로젝트인 ARISTO1은 4, 8, 12학년 수준에서 점점 더 어려워지는 과학 시험에서 파생된 QA 작업 모음을 중심으로 구성됩니다. Richardson et al. (2013)은 MCTest2에 660개의 이야기 세트와 텍스트의 기계 이해에 대한 연구를 위한 관련 질문을 제안했습니다. 각 질문은 독자가 이야기의 다양한 측면을 이해하도록 요구합니다.

이 두 이니셔티브는 유망한 방향으로 진행되지만 이러한 벤치마크에 대한 결과를 해석하는 것은 여전히 복잡합니다. 실제로 제안된 작업을 완전히 해결할 수 있는 시스템은 아직 없으며 질문에 답하기 위해 많은 하위 작업을 해결해야 하기 때문에(대역, 연역, 상식 사용 등) 명확하게 설명하기 어렵습니다. 이러한 시스템의 기능과 한계를 식별하여 개선 및 수정을 제안합니다. 결과적으로 이러한 프로젝트에서 도출된 결론은 대규모 지식 기반에 대한 QA에 대한 보다 전통적인 작업에서 도출된 결론보다 훨씬 명확하지 않습니다(Berant et al., 2013; Fader et al., 2014). 게다가 최고의 성능을 발휘하는 시스템은 수작업으로 만든 패턴과 특징 및/또는 매우 큰 말뭉치에서 얻은 통계를 기반으로 합니다. 그러한 시스템이 실제로 언어를 이해하고 단순히 전통적인 정보 추출 방법의 가벼운 업그레이드가 아니라고 주장하기는 어렵습니다(Yao et al., 2014). Berant 등의 시스템(2014)은 텍스트와 대답할 질문의 구조화된 표현을 구축하기 때문에 더욱 발전했습니다. 그 잠재력에도 불구하고 이 방법은 고도로 도메인 특정적이며 많은 사전 지식에 의존합니다.

이러한 관찰을 바탕으로 우리는 시스템의 실패 또는 성공이 해당 기능에 대한 피드백을 명확하게 제공할 수 있다는 주요 목표와 함께 훨씬 더 간단한 QA 작업 모음을 구성하기로 했습니다. 그 점에서 우리는 Winograd Schema Challenge Levesque et al. (2011), 다음과 같은 단일 이항 선택 질문이 뒤따르는 간단한 문장으로 구성되어 있습니다. “Joan은 Susan이 받은 모든 도움에 대해 Susan에게 감사를 표했습니다. 누가 도움을 받았습니까? 조안이나 수잔?”. 이 도전과 우리의 임무에서 결과를 해석하는 것은 간단합니다.

그러나 Winograd Challenge는 대부분 시스템이 성명서에 표현되지 않은 배경 지식을 획득하고 사용할 수 있는지 평가하는 데 중점을 두지만 우리의 작업은 독립적이고 더 다양합니다. 독립적이라는 것은 ARISTO 및 Winograd Challenge의 경우처럼 후자만이 아니라 교육 데이터와 평가 데이터가 모두 포함된 작업을 의미합니다. MCTest에는 훈련/테스트 분할이 있지만 훈련 세트가 너무 작아서 테스트 세트에서 잘 수행하는 데 필요한 모든 추론을 캡처할 수 없습니다. 설정에서 잘 수행하는 데 필요한 교육 예제의 양(원하는 만큼 늘릴 수 있음)을 평가할 수 있으며 테스트 세트에 필요한 상식과 추론이 교육 세트에 포함되어야 합니다. 다양성 측면에서 우리 작업 중 일부는 기존 설정과 관련이 있지만 많은 추가 작업도 제안합니다. 작업 8과 9는 예를 들어 람다 종속성 기반 구성 의미론(Liang et al., 2013; Liang, 2013)에 대한 이전 작업에서 영감을 받았습니다. 우리에게 각 작업은 시스템이 반드시 수행해야 하는 하나의 기술을 확인합니다.

¹ <http://allenai.org/aristo.html> ² <http://research.microsoft.com/mctest>

우리는 전체 텍스트 이해 및 추론을 목표로 하는 모든 시스템의 전제 조건으로 모든 시스템에서 잘 수행하는 것을 가정합니다.

3 과제 _

원칙 우리의 주요 아이디어는 컴퓨터 과학에서 소프트웨어 테스팅이 구축되는 방식과 유사한 방식으로 일련의 작업을 제공하는 것입니다. 이상적으로 각 작업은 가능한 한 다른 작업과 독립적인 "리프트" 테스트 사례이며 의도된 동작의 가능한 한 측면을 가장 간단한 방법으로 테스트합니다. 후속("비-리프트") 테스트는 조합을 테스트하여 이들을 기반으로 구축할 수도 있습니다. 작업은 <http://fb.ai/babi>에서 공개적으로 사용할 수 있습니다. 작업을 생성하는 소스 코드는 <https://github.com/facebook/bAbi-tasks>에서 사용할 수 있습니다.

각 작업은 성공적인 모델이 테스트 데이터에서 잘 수행되도록 교육 및 테스트 데이터 세트를 제공합니다. Weston et al. (2014), 훈련 세트의 감독은 질문에 대한 진정한 답변과 학습자가 사용할 수도 있고 사용하지 않을 수도 있는 주어진 질문에 대답하기 위한 관련 진술 세트에 의해 제공됩니다. 우리는 정답이 한 단어 (Q: Mark는 어디에 있습니까? A: 화장실) 또는 단어 목록 (Q: Mark는 무엇을 들고 있습니까?) 로 제한되도록 작업을 설정했습니다. 그러면 평가가 명확해집니다. 그리고 단순히 옳고 그름으로 측정됩니다.

모든 작업은 소음이 없으며 해당 언어를 읽을 수 있는 사람은 잠재적으로 100% 정확도를 달성할 수 있습니다. 우리는 인간에게 자연스러운 작업을 선택하려고 노력했습니다. 작업은 단순하고 일상적인 상황을 기반으로 하며 성인이 문제를 해결하는 데 필요한 형식적 의미론, 기계 학습, 논리 또는 지식 표현과 같은 영역에 대한 배경 지식이 없습니다.

데이터 자체는 섹션 4에 설명된 위치에서 이동하고 상호 작용하는 캐릭터와 개체의 간단한 시뮬레이션을 사용하여 생성됩니다. 시뮬레이션을 통해 실제 레이블이 시뮬레이션에 기반하여 알려진 다양한 시나리오에서 데이터를 생성할 수 있습니다. 각 작업에 대해 표 1과 2의 진술, 질문 및 실제 레이블(빨간색)을 포함하는 데이터 세트의 작은 샘플을 제공하여 설명합니다.

단일 지원 사실 작업 1은 이전에 제공된 단일 지원 사실(잠재적으로 다른 관련 없는 사실 세트 중에서)이 답변을 제공하는 질문으로 구성됩니다. 우리는 먼저 "Mary가 사무실에 갔다. 메리는 어디에 있습니까?". 이러한 종류의 작업은 이미 Weston et al. (2014). Fader et al.과 같은 일부 실제 QA 데이터 세트의 가장 간단한 사례로 간주될 수 있습니다. (2013).

2개 또는 3개의 지원 사실 더 어려운 작업은 질문에 답하기 위해 두 개의 지원 진술을 연결해야 하는 질문에 답하는 것입니다. 하나는 "John is in the playground" 와 "John pick up the football" 이라는 두 문장의 정보를 결합해야 합니다. 다시 말하지만, 이러한 종류의 작업은 이미 Weston et al. (2014).

유사하게, 과제 3에 주어진 세 가지 지원 사실로 과제를 만들 수 있습니다. 여기서 처음 세 문장은 모두 "주방 앞에 사과가 어디에 있었나요?"라는 질문에 답해야 합니다.

두 개 또는 세 개의 인수 관계 질문에 답하려면 주제와 대상을 구별하고 인식하는 능력이 중요합니다. 작업 4에서 우리는 문장이 재정렬된 단어를 특징으로 하는 극단적인 경우, 즉 bag-of-words가 작동하지 않는 경우를 고려합니다. 예를 들어 "침실 북쪽에 무엇이 있습니까?"라는 질문이 있습니다. 그리고 "북쪽 침실은 무엇입니까?" 똑같은 단어지만 순서가 다르고 대답도 다릅니다. 한 단계 더 나아가 때로는 세 가지 별도의 주장을 구별해야 합니다. 작업 5는 "Jeff는 Bill에 의해 우유를 받았습니까"와 같은 진술을 포함 하고 누가 제공자, 수신자 또는 관련 대상이 누구인지 질문합니다.

예/아니오 질문 작업 6은 가능한 가장 간단한 질문 중 일부(구체적으로 단일 지원 사실이 있는 질문)에 대해 "존이 놀이터에 있습니까?" 와 같은 참/거짓 유형 질문에 답하는 모델의 능력을 테스트합니다.

계산 및 나열/설정 작업 7은 "다니엘이 들고 있는 물건이 몇 개인가?" 와 같이 특정 속성을 가진 물건의 수에 대해 질문하여 간단한 계산 작업을 수행하는 QA 시스템의 능력을 테스트합니다. 유사하게, 작업 8은 "Daniel이 들고 있는 것이 무엇입니까?" 와 같이 목록 형식으로 한 단어로 된 답변 세트를 생성하는 능력을 테스트합니다. 이러한 작업은 기본 데이터베이스 검색 작업과 관련된 QA 작업으로 볼 수 있습니다.

표 1: 작업 1에서 10까지의 샘플 진술 및 질문.

작업 1: 단일 지원 사실 Mary는 화장실에 갔다. John은 복도로 이동했습니다. 메리는 사무실로 갔다. 메리는 어디에 있습니까? A: 사무실	작업 2: 두 가지 지원 사실 John은 놀이터에 있습니다. 존은 축구공을 집어 들었다. 밥은 부엌으로 갔다. 축구는 어디에 있습니까? 놀이터
작업 3: 세 가지 지원 사실 John이 사과를 집어 들었습니다. 존은 사무실로 갔다. 존은 부엌으로 갔다. 존은 사과를 떨어뜨렸다. 부엌 앞의 사과는 어디에 있었습니까? A: 사무실	작업 4: 두 인수 관계 사무실은 침실 북쪽에 있습니다. 침실은 욕실 북쪽에 있습니다. 부엌은 정원의 서쪽에 있습니다. 침실 북쪽은 무엇입니까? A: 사무실 북쪽 침실은 어디입니까? 화장실
과제 5: 세 가지 인수 관계 Mary는 Fred에게 케이크를 주었습니다. Fred는 Bill에게 케이크를 주었다. Jeff는 Bill에게 우유를 받았습니다. 누가 프레드에게 케이크를 줬나요? A: Mary Fred는 누구에게 케이크를 주었습니까? A: 빌	작업 6: 예/아니오 질문 John은 놀이터로 이동했습니다. 다니엘은 화장실에 갔다. 존은 복도로 돌아갔다. John은 놀이터에 있습니까? A: 아니 다니엘이 화장실에 있어? A: 예
과제 7: Daniel이 축구공을 집었습니다. 다니엘은 축구공을 떨어뜨렸습니다. 다니엘은 우유를 얻었습니다. 다니엘은 사과를 가져갔습니다. 다니엘이 들고 있는 물건은 몇 개입니까? 답: 둘	작업 8: 목록/세트 다니엘이 축구공을 집습니다. 다니엘이 신문을 떨어뜨립니다. 다니엘이 우유를 집었습니다. 존은 사과를 가져갔다. 다니엘이 들고 있는 것은? 우유, 축구
작업 9: 단순 부정 산드라는 사무실로 갔다. Fred는 더 이상 사무실에 없습니다. 프레드는 사무실에 있나요? A: 아니 오 Sandra는 사무실에 있나요? A: 예	작업 10: 무한한 지식 John은 교실이나 운동장에 있습니다. 산드라는 정원에 있습니다. John은 교실에 있습니까? A: 아쩌면 John은 사무실에 있나요? A: 아니요

단순 부정 및 무기한 지식 작업 9 및 10은 약간 더 복잡한 자연 언어 구조를 테스트합니다. 작업 9는 부정의 가장 단순한 형태 중 하나인 "Fred가 사무실로 이동했습니다"가 아니라 "Fred는 더 이상 사무실에 없습니다"와 같이 진술이 거짓임을 암시하는 지원 사실을 테스트합니다. (이 경우 작업 6(예/아니오 질문)은 작업의 전제 조건입니다.) 작업 10은 확실성보다는 가능성을 설명하는 진술을 모델링할 수 있는지 테스트합니다. 예를 들어 "John은 교실이나 운동장에 있습니다.", 이 경우 "John이 교실에 있습니까?"라는 질문에 대한 대답은 "아마도"입니다.

기본 상호 참조, 접속사 및 복합 상호 참조 작업 11은 가장 간단한 유형의 상호 참조를 테스트합니다. 예를 들어 "Daniel was in the kitchen. 그런 다음 그는 스튜디오에 갔다.". 실제 데이터는 일반적으로 이를 라벨링 문제로 다루고 보다 정교한 현상을 연구하는 반면(Soon et al., 2001), 우리는 다른 모든 작업에서 질문 응답 문제로 평가합니다. 작업 12(접속사) 테스트는 "Mary와 Jeff가 부엌에 갔다."와 같이 하나의 진술에서 여러 주제를 언급합니다. 작업 13 테스트는 대명사가 여러 행위자를 지칭할 수 있는 경우(예: "Daniel과 Sandra가 사무실로 이동했습니다. 그런 다음 그들은 정원으로 갔다.")

시간 추리 지금까지 우리의 과제가 진술 순서에 시간을 암시적으로 포함하는 동안, 과제 14는 진술 내에서 시간 표현의 사용을 이해하는 테스트입니다. 예를 들어 "오후에 Julie는 공원에 갔다. 어제 Julie는 학교에 있었다". 실제 데이터 세트는 일반적으로 QA 작업이 아닌 라벨링으로 시간 표현을 평가하는 작업을 처리합니다(예: UzZaman et al. 참조). (2012).

기본 공제 및 귀납 과제 15는 속성 상속을 통한 기본 공제를 테스트합니다.

"양은 늑대를 무서워합니다. 거트루드는 양입니다. Gertrude는 무엇을 두려워합니까?". 마찬가지로 작업 16

표 2: 작업 11~20의 샘플 진술 및 질문.

작업 11: 기본 공동 참조 다니엘은 부엌에 있었다. 그런 다음 그는 스튜디오로 갔다. 산드라는 사무실에 있었다. 다니엘은 어디에 있습니까? A: 스튜디오	작업 12: Conjunction Mary 와 Jeff는 부엌으로 갔다. 그런 다음 Jeff는 공원에 갔다. 메리는 어디에 있습니까? A: 부엌 Jeff 는 어디 있니? 공원
작업 13: 복합 공동 참조 Daniel과 Sandra는 사무실로 이동했습니다. 그런 다음 그들은 정원으로 갔다. 산드라와 존은 부엌으로 갔다. 그 후 그들은 복도로 이동했습니다. 다니엘은 어디에 있습니까? 정원	작업 14: 시간 추론 오후에 Julie는 공원에 갔다. 어제 Julie는 학교에 있었다. Julie는 오늘 저녁에 영화관에 갔다. Julie는 공원 후에 어디로 갔습니까? A: 시네마 줄리는 공원 앞 에 어디 있었어? 학교
과제 15: 기본 공제 양은 늑대를 무서워합니다. 고양이는 개를 무서워합니다. 쥐는 고양이를 무서워합니다. 거트루드는 양입니다. 거트루드는 무엇을 두려워합니까? A: 늑대	작업 16: 기본 귀납법 릴리는 백조입니다. 릴리는 흰색입니다. Bernhard는 녹색입니다. 그렉은 백조입니다. 그렉은 무슨 색인가요? 흰색
작업 17: 위치 추론 삼각형은 파란색 사각형 의 오른쪽에 있습니다. 빨간색 사각형은 파란색 사각형 위에 있습니다. 빨간색 구는 파란색 사각형의 오른쪽에 있습니다. 파란색 사각형 오른쪽에 빨간색 구체가 있습니까? A: 예 삼각형의 왼쪽에 빨간색 사각형이 있습니까? A: 예	작업 18: 크기 추론 축구공은 가방에 쏙 들어갑니다. 여행 가방은 찬장에 맞습니다. 상자는 축구공보다 작습니다. 상자가 여행 가방에 들어갈까요? A: 예 찬장이 상자 에 맞습니까? A: 아니요
작업 19: 길 찾기 주방은 복도 북 쪽에 있습니다. 욕실은 침실 서쪽에 있습니다. 굴은 복도 동쪽에 있습니다. 사무실은 침실 남쪽에 있습니다. 서재에서 부엌으로 어떻게 가나요? A: 서쪽, 북쪽 사무실에서 화장실까 지 어떻게 가나요? A: 북쪽, 서쪽	작업 20: 에이전트의 동기 John은 배가 고 프다. 존은 부엌으로 간다. 존은 그곳에서 사과를 집어 들었다. 다니엘은 배가 고프다. 다니엘은 어디로 갑니까? A: 부엌 존은 왜 부엌 에 갔어? A: 배고프다

속성 상속을 통해 기본 유도를 테스트합니다. 귀납과 연역에 대한 완전한 분석은 분명히 이 작업의 범위를 벗어나며, 향후 작업은 더 심도 있는 측면을 분석해야 합니다.

위치 및 크기 추론 작업 17은 색 블록의 상대적 위치에 대한 질문을 통해 고전적인 SHRDLU 시스템(Winograd, 1972)의 많은 구성 요소 중 하나인 공간 추론을 테스트합니다. 작업 18은 개체의 상대적 크기에 대한 추론을 필요로 하며 Winograd 스키마 챌린지(Levesque et al., 2011)의 상식적인 추론 예제에서 영감을 받았습니다.

경로 찾기 작업 19의 목표는 위치 사이의 경로를 찾는 것입니다. 다양한 위치에 대한 설명이 주어지면 다음과 같이 질문합니다. 한 위치에서 다른 위치로 이동하는 방법은 무엇입니까? 이는 Chen & Mooney(2011)의 작업과 관련이 있으며 효과적으로 검색 문제를 포함합니다.

Agent's Motivations 마지막으로 가능한 가장 간단한 방법으로 에이전트가 행동을 수행하는 이유에 대한 20개의 질문을 과업으로 합니다. 배우가 주어진 상태(배고프고, 목마르고, 피곤하고...)에 있는 경우와 배우가 취한 조치(예: 배고픈 사람이 부엌에 갈 수 있다는 것을 학습해야 함)를 다룹니다.

이미 언급했듯이 이러한 작업은 기계 학습 알고리즘의 개발 및 이해를 촉진하기 위한 것입니다. 모든 작업에서 단일 모델을 평가해야 하며(작업당 조정 아님) 동일한 모델을 추가 실제 작업에서 테스트해야 합니다.

데이터 릴리스에서는 위의 20개 작업을 영어로 제공하는 것 외에도 (i) 힌디어로 제공합니다. (ii) 사람이 더 이상 읽을 수 없도록 영어 단어를 섞습니다. 좋은 학습 알고리즘은 세 가지 모두에서 유사하게 수행되어야 합니다. 외부 리소스를 사용하는 방법의 경우에는 그렇지 않을 수 있습니다. 이러한 설정은 처음에 언어를 제시하고 처음부터 학습해야 하는 학습자를 모방하기 위한 것입니다.

4 시뮬레이션

우리의 모든 작업은 고전적인 텍스트 어드벤처 게임처럼 작동하는 시뮬레이션으로 생성됩니다. 아이디어는 이 시뮬레이션 내에서 텍스트를 생성하면 사용되는 언어를 일관되고 통제된(인공) 세계에 정착시킬 수 있다는 것입니다. 우리의 시뮬레이션은 Bordes 등의 시뮬레이션을 따릅니다. (2010); Weston et al. (2014) 그러나 다소 더 복잡합니다.

시뮬레이션된 세계는 다양한 유형(위치, 개체, 사람 등)의 엔터티와 이러한 엔터티에서 작동하는 다양한 작업으로 구성됩니다. 엔터티에는 내부 상태가 있습니다. 즉, 위치, 개체가 위에 있는지 내부에 있는지 여부(예: 테이블 및 상자), 행위자의 정신 상태(예: 배고픈 상태), 크기, 색상 및 식용 가능성과 같은 속성이 있습니다. 위치의 경우 연결된 인근 장소(예: 동쪽 또는 위에 있는 것)가 인코딩됩니다. 액터의 경우 액터별로 사전 지정된 규칙 세트를 지정하여 행동을 제어할 수도 있습니다. 예를 들어 배고프면 음식을 찾으려고 할 수 있습니다. 규칙이 설정되지 않은 경우 임의의 유효한 작업을 실행할 수도 있습니다(예: 무작위로 걸어 다니기).

액터가 시뮬레이션에서 실행할 수 있는 작업은 다음과 같이 구성됩니다. `_ _ 행위자`, `<객체>` 놓기, `<엔터티>` `<상태>` 설정, 보기, 인벤토리 및 `<객체>` 검사. 시뮬레이션에서 일관성을 유지하기 위해 이러한 작업에 일련의 보편적 제약이 부과됩니다. 예를 들어 액터는 자신이나 다른 사람이 이미 가지고 있는 것을 얻을 수 없고, 현재 위치와 연결되지 않은 장소로 갈 수 없으며, 아직 가지고 있지 않은 것을 떨어뜨릴 수 없습니다. 기본 작업, 액터에 대한 규칙 및 해당 제약 조건을 사용하여 액터가 작동하는 방식을 정의합니다. 각 작업에 대해 해당 작업에 필요한 작업을 제한합니다. 예를 들어 작업 1은 go 만 필요한 반면 작업 2는 go, get 및 drop을 사용합니다. 명령을 적어 놓으면 시뮬레이션으로 실행할 수 있는 매우 간단한 "이야기"가 제공됩니다(예: joe go 놀이터; 밥은 사무실에 간다; 조는 축구를 얻습니다. 이 예는 작업 2에 해당합니다. 그런 다음 시스템은 시뮬레이션 상태에 대해 질문할 수 있습니다(예: where john?, where football?). 등등. 기본 세계에 접근할 수 있으므로 이러한 질문에 대한 진정한 답을 쉽게 계산할 수 있습니다.

문장과 질문에서 다양한 어휘로 보다 자연스럽게 보이는 텍스트를 생성하기 위해 간단한 자동 문법을 사용합니다. 각 동사에는 일련의 동의어가 지정됩니다. 예를 들어, 시뮬레이션 명령 get 은 즐기, 가져오기, 움켜쥐기 또는 가져오기 로 대체되고 드롭 은 떨어뜨리기, 왼쪽, 버리기 또는 놓기 로 대체됩니다. 마찬가지로, 각 개체와 행위자는 대체 동의어 집합을 가질 수 있습니다. 예를 들어 Daniel을 작업 11에서 he 로 교체합니다. 부사는 시간 추론 작업 14와 같은 일부 작업에 중요합니다.

아직 모델링되지 않은 언어의 많은 측면이 있습니다. 예를 들어, 모든 문장은 지금까지 상대적으로 짧고 중첩이 거의 없습니다. 또한 엔터티와 어휘 크기가 작습니다(150단어, 일반적으로 작업당 사용되는 4명의 행위자, 6개의 위치 및 3개의 객체). 잘 정의된 일련의 작업을 정의하면 실제 데이터로는 어려운 시뮬레이션 환경 내에서 제어된 방식으로 모델을 평가하는 데 도움이 될 것입니다. 즉, 이러한 작업은 실제 데이터를 대체하는 것이 아니라 특히 알고리즘을 개발하고 분석할 때 이를 보완해야 합니다.

5 실험

우리는 (영어 데이터 세트에서) 작업에 대해 다음 방법을 비교했습니다. (i) N-그램 분류기 기준선, (ii) LSTM(장단기 기억 순환 신경망) (Hochreiter & Schmidhuber, 1997), (iii) 메모리 네트워크(MemNN)(Weston et al., 2014), (iv) 메모리 네트워크의 일부 확장에 대해 자세히 설명하겠습니다. (v) 기존 NLP 작업의 외부 레이블 데이터를 통합하는 구조화된 SVM. 이러한 모델은 세 개의 개별 트랙에 속합니다. 약하게 감독된 모델은 교육 시간에 질문 답변 쌍만 제공되는 반면 강력한 감독은 교육 시간(테스트 시간이 아님)에도 일련의 지원 사실을 제공합니다. 강하게 감독된 모델은 약한 감독 모델에 대한 정확도 상한을 제공합니다. 즉, 동일한 모델 클래스에서 성능이 우수해야 합니다. 마지막 외부 리소스 트랙의 방법은 제공된 훈련 세트가 아닌 다른 소스의 레이블이 지정된 데이터(예: 상호 참조 및 의미론적 역할 레이블 작업, 강력한 감독)를 사용할 수 있습니다. 각 작업에 대해 교육용으로 1000개의 질문을 사용하고 테스트용으로 1000개의 질문을 사용하고 테스트 정확도를 보고합니다. 95% 이상의 정확도를 얻으면 작업을 성공적으로 통과한 것으로 간주합니다 3.

395%(및 1000개의 훈련 예제)의 선택은 임의적입니다.

표 3: 다양한 방법에 대한 20가지 작업에 대한 테스트 정확도(%) (각각 1000개의 교육 예제). 우리가 제안하는 MemNN 확장은 열 5-9에 있습니다: 적응형 메모리(AM), N-그램(NG), 비선형 일치 함수(NL) 및 이들의 조합. 굵은 숫자는 확장이 95% 이상의 정확도를 달성하지만 Weston 등의 원래 MemNN 모델이 달성하는 작업을 나타냅니다. (2014) 하지 않습니다. 마지막 두 열(10-11)은 MemNN 방법에 대한 추가 분석을 제공합니다. 열 10은 각 작업에 필요한 교육 데이터의 양을 제공합니다.

오전 + NG + NL

95% 이상의 정확도를 얻거나 1000개의 훈련 예제로 달성할 수 없는 경우 실패합니다. 마지막 열은 개별적으로가 아니라 한 번에 모든 데이터를 훈련할 때 정확도를 제공합니다.

일	약하게 감독		외부 사용 자원	강력한 감독(자원 사실 사 용)				
	W1S1	W1S2	W1S3	MemNN	MemNN	MemNN	MemNN	MemNN
1 - 단일 뒷받침 사실 2 - 두 가지 뒷받침 사실 3 - 세 가지 뒷받침 사실 4 - 두 인수. 관계 5 - 세 인수. 관계 6 - 예/아니오 질문 7 - 세기 8 - 목록/집합 9 - 단순 부정 10 - 부정확한 지식 11 - 기본 대용어 12 - 접속사 13 - 복합 코어프.	36	50	99	100	100	100	100	예 100 250 100 100 500
14 - 시간 추론 15 - 기본 추론 16 - 기본 귀납 17 - 위치 추론 18 - 크기 추론 19 - 경로 찾기 20 - 행위자의 동기 평균 성과	2	20	74	100	100	100	100	예. 100 100 500 예. 98 100
	7	20	17	20	100	99	100	500 예. 80 98 1000 예. 99
	50	61	98	71	69	100	73	100 500 예. 100 85 실패 86
	20	70	83	83	83	86	86	91 실패 93 100 500 예. 100
	49	48	99	47	52	53	100	98 1000 예. 98 100 250 예.
	52	49	69	68	78	86	83	100 100 250 예. 100 1000
	40	45	70	77	90	88	94	예. 100 99 500 예. 99 1000
	62	64	100	65	71	63	100	예. 100 100 100 예. 94 65 실패
	45	44	99	59	57	54	97	패 72 95 1000 예. 93 36 실패
	29	72	100	100	100	100	100	패 19 100 250 ex. 100 93
	9	74	96	100	100	100	100	92
	26	94	99	100	100	100	100	
	19	27	99	99	100	99	100	
	20	21	96	74	73	100	77	
	43	23	24	27	100	100	100	
	46	51	61	54	46	49	57	
	52	52	62	57	50	74	54	
	0	8	49	0	9	3	15	
	76	91	95	100	100	100	100	
	34	49	79	75	79	83	87	

방법 N-그램 분류기 기준선은 Richardson et al. (2013) 그러나 객관식 질문이 아닌 1단어 답변을 생성하는 경우에 적용: 이야기에서 질문과 적어도 하나의 단어를 공유하는 모든 문장에 대해 bag-of-N-grams를 구성한 다음 이러한 기능을 사용하여 답을 예측하는 선형 분류기를 학습합니다 4.

LSTM은 시퀀스 예측에 널리 사용되는 방법이며(Sutskever et al., 2014) 우리와 유사한 작업에서 표준 RNN(Recurrent Neural Networks)을 능가합니다(Weston et al., 2014). 그들은 질문에 도달할 때까지 이야기를 읽은 다음 답을 출력해야 합니다. 답변만으로는 약하게 감독되므로 강력한 감독 방법이나 외부 리소스를 사용하는 방법에 비해 불리합니다.

MemNN(Weston et al., 2014)은 QA에서 잘 수행되는 것으로 나타난 최근에 제안된 모델 클래스입니다. 그것들은 스토리의 이전 진술로 구성된 저장된 기억에 대해 추론을 수행하는 "컨트롤러" 신경망에 의해 작동합니다. 원래 제안된 모델은 2개의 추론 홉을 수행합니다. 즉, 질문과 최대 일치 점수가 있는 첫 번째 지원 사실을 찾은 다음 질문과 발견된 첫 번째 사실 모두와 최대 일치 점수가 있는 두 번째 지원 사실을 찾습니다. 매칭 기능은 단어 임베딩을 합산하여 질문과 사실에 대한 단어 모음을 임베딩 공간에 매핑하는 것으로 구성됩니다. 단어 임베딩은 QA 작업을 최적화하기 위해 강력한 감독을 사용하여 학습됩니다. 지원 사실을 찾은 후, 해당 사실에 대해 가능한 응답(답변 단어)의 순위를 매기기 위해 최종 순위가 수행됩니다. 또한 이 모델에 대한 몇 가지 확장을 고려합니다.

- 2가 아닌 다양한 수의 홉을 수행하는 적응형 메모리 모델은 홉 또는 특수 "STOP" 클래스를 예측하도록 훈련됩니다. 유사한 절차를 적용하여 여러 토큰을 출력할 수도 있습니다.

4 필터링된 집합을 사용하는 대신 모든 문장에서 N-gram을 구성하면 더 나쁜 결과가 나타납니다.

- N-grams 우리는 텍스트를 표현하기 위해 bag-of-words가 아닌 bag of 3-grams를 사용해 보았습니다. 두 경우 모두 MemNN의 첫 번째 단계는 이를 벡터 임베딩으로 변환하는 것입니다.
- 비선형성 우리는 tanh 비선형성을 가진 고전적인 2층 신경망을 다음과 같이 적용합니다. 매칭 가능.

이러한 변형에 대한 자세한 내용은 부록의 섹션 A에 나와 있습니다.

마지막으로 우리는 구조화된 SVM(Support Vector Machine)을 사용하여 고전적인 캐스케이드 NLP 시스템 기준선을 구축했습니다. 이 시스템은 상호 참조 해상도 및 SRL(semantic role labeling) 전처리 단계를 통합하며 비용이 많이 드는 대량의 레이블이 지정된 데이터에 대해 자체적으로 훈련됩니다. Stanford 공동 참조 시스템(Raghunathan et al., 2010)과 SENNA semantic role labeling(SRL) 시스템(Collobert et al., 2011)은 SVM에 대한 입력을 위한 기능을 구축하는 데 사용되며 강력한 슈퍼 비전으로 훈련되어 지원 사실, 예: 단어, 단어 쌍, SRL 동사 및 동사-인수 쌍에 기반한 기능. 지원 사실을 찾은 후 해당 목표에 맞게 조정된 기능을 사용하여 응답 단계에 대해 유사한 구조화된 SVM을 구축합니다. 자세한 내용은 초에 있습니다. 부록 B.

모든 방법에 대한 학습률 및 기타 하이퍼파라미터는 훈련 세트를 사용하여 선택됩니다. 작업에 대한 실험 결과 요약은 표 3에 나와 있습니다. 20개 작업 각각에 대한 결과와 마지막 두 행에 평균 성능 및 실패한 작업 수를 제공합니다.

결과 표준 MemNN은 일반적으로 N-gram 및 LSTM 기준선을 능가하며 이는 Weston et al.의 결과와 일치합니다. (2014). 그러나 그들은 여전히 많은 작업에서 "실패"합니다. 즉, 테스트 정확도는 95% 미만입니다. 이러한 실패 중 일부는 Sec. A.1, 예를 들어 $k=2$ 사실, 단일 단어 답변 및 단어 모음은 작업 3, 4, 5, 7, 8 및 18에서 성공하지 못했습니다. 그러나 처음에는 예상하지 못한 작업에서도 실패했습니다. 예를 들어 예/아니오 질문(6) 및 불명확한 지식(10). 돌이켜보면 표준 MemNN의 선형 채점 기능은 3자 상호작용이 필요하기 때문에 쿼리, 지원 사실 및 예/아니오 대답 간의 일치치를 모델링할 수 없다는 것을 깨달았습니다.

표 3의 5-9열은 MemNN 확장에 대한 결과를 제공합니다: 적응형 메모리(AM), N-그램(NG) 및 비선형성(NL) 및 이들의 조합. 적응형 접근 방식은 작업 3과 16에서 두 개 이상의 지원 사실이 필요하기 때문에 직접적인 개선을 제공하고, 8과 19에서는 여러 단어 출력이 필요하기 때문에 (하지만 여전히 어렵습니다) (작은) 개선을 제공합니다. 따라서 후속 실험에서 다른 모든 확장과 함께 AM 모델을 사용합니다.

N-gram 모델링을 사용하는 MemNN은 단어 순서가 중요한 경우(예: 작업 4 및 15) 분명한 개선을 제공합니다. 앞에서 설명한 대로 yes/no(6) 및 indefinite 작업(10)에서. 반면에 NL 방법은 단어 순서를 모델링할 수 없으므로 예를 들어 작업 4에서 실패합니다. 따라서 분명한 단계는 이러한 보완적인 접근 방식을 결합하는 것입니다. 원래 MemNN 모델에 비해 실패에서 성공으로 업그레이드된 총 9개의 작업.

구조화된 SVM은 외부 리소스에 액세스할 수 있음에도 불구하고 더 나은 성능을 발휘하지 못하며 여전히 9개의 작업에서 실패합니다. 손으로 만든 기능 연결이 필요한 비선형성을 캡처하는 작업 6, 9 및 10에서 바닐라 MemNN(확장 없음)보다 성능이 좋습니다. 그러나 MemNN(AM+NG+NL)과 비교할 때 3개(때로는 2개)의 지원 사실(예: 작업 3, 16 및 2)이 필요한 작업에서는 훨씬 더 나쁜 것으로 보입니다. 실수. 그러나 탐욕스럽지 않은 검색은 검색이 매우 중요한 경로 찾기(작업 19)와 같은 다른 작업에 도움이 되는 것 같습니다. 영어용으로 특별히 설계된 외부 리소스에 의존하기 때문에 이러한 외부 리소스의 품질이 더 나쁠 수 있는 힌디어와 같은 다른 언어에서도 제대로 작동하지 확인할 수 없습니다.

마지막 두 열(10-11)은 AM+NG+NL MemNN 방법에 대한 추가 분석을 제공합니다. 두 번째에서 마지막 열(10)은 $\geq 95\%$ 정확도를 달성하는 데 필요한 훈련 예제의 최소 수를 보여줍니다. 또는 1000개의 예제로 달성하지 못한 경우 실패합니다. 이것은 작업을 잘 수행하는 것이 바람직할 뿐만 아니라 가장 적은 수의 예제를 사용하는 것(잘 일반화하기 위해)도 중요하기 때문에 중요합니다. 대부분의 후속 작업에는 100-500개의 예제가 필요합니다. 작업 8에는 5000개의 예제가 필요하고 7에는 10000개의 예제가 필요하므로 실패로 레이블이 지정됩니다. 후자의 작업은 물건을 집을 시간을 모두 더하고 물건을 떨어뜨리는 시간을 빼면 해결될 수 있을 것 같습니다.

MemNN의 경우 가능하지만 완벽하게 수행되지는 않습니다. 두 가지 작업, 위치 추론 17 및 경로 찾기 19는 10000개의 예로도 해결할 수 없습니다. 이러한 작업(실제로 우리가 구축할 예정인 더 발전된 형태의 귀납 및 연역)은 추론 절차에 내장된 일반 검색 알고리즘이 필요한 것 같습니다., MemNN (및 시도한 다른 접근 방식)이 부족합니다.

마지막 열은 단일 작업이 아닌 모든 작업을 공동으로 교육할 때 AM+NG+NL MemNN의 성능을 보여줍니다. 성능은 일반적으로 고무적으로 유사하며, 이러한 모델이 텍스트 이해 및 추론의 여러 측면을 동시에 학습할 수 있음을 보여줍니다. 주요 문제는 이러한 모델이 여전히 여러 작업에서 실패하고 일반적으로 현실적인 것보다 훨씬 더 강력한 형태의 감독(지원 사실 사용)을 사용한다는 것입니다.

6 토론

전제 조건 세트 우리는 완전한 언어 이해 및 추론을 위한 전제 조건이라고 믿는 일련의 작업을 개발했습니다. 이러한 작업을 해결할 수 있는 학습자가 반드시 완전한 추론에 가까운 것은 아니지만, 학습자가 우리의 작업 중 하나라도 실패하면 실패할 실제 작업이 있을 수 있습니다(즉, 동일한 작업을 요구하는 실제 작업). 일종의 추론). 작업의 상황과 언어가 인위적일지라도 이를 해결하는 방법을 배우는 데 필요한 메커니즘이 텍스트 이해 및 추론의 핵심 부분이라고 믿습니다.

유연한 프레임워크 이 작업 집합은 결정적인 집합이 아닙니다. 시뮬레이션 기반 접근 방식의 목적은 작업 구성의 유연성과 제어를 제공하는 것입니다. 작업의 유용성을 이해하고 결과를 해석하는 것이 더 쉽기 때문에 우리는 작업을 언어에 근거했습니다. 그러나 우리의 주요 목표는 기초 시퀀스에서 패턴을 감지하고 결합하는 방법을 학습할 수 있는 모델을 찾는 것입니다. 언어적 의미에서 제거된 단순한 기초에 대해서만 어휘 가변성과 모호성 및 이유를 제거하여 본질적인 어려움을 줄이고자 할 수도 있습니다. 예를 들어 지원 사실을 질문에 더 가깝게 배열하여 시스템의 추론 기능에서 장기 기억을 분리할 수도 있습니다. 반대 관점에서 주석이나 더 복잡한 문법을 사용하여 작업을 보다 현실적인 이야기로 변환할 수 있습니다. 여기에 제시된 20개의 작업 세트는 시뮬레이션으로 달성할 수 있는 것의 하위 집합입니다. 우리는 텍스트 추론 모델이 갖기를 바라는 다양한 기술을 제공하기 때문에 그것들을 선택했지만, 커뮤니티의 연구자들이 이를 해결하려는 모델을 개발하고 분석하기 위해 다양한 복잡성의 더 많은 작업을 개발하기를 바랍니다. 작업 간 전이 학습도 이 백서의 범위를 벗어나는 매우 중요한 목표입니다. 따라서 우리는 이러한 목적을 위해 공개적으로 사용할 수 있는 작업에 대한 시뮬레이터와 코드를 만들었습니다.

테스트 학습 방법 우리의 과제는 학습 방법을 위한 시험대로 설계되었습니다. 우리는 모델 안에 숨겨진 패턴에서 추론하는 방법을 발견하기 위해 모델의 기능을 평가하기 위해 교육 및 테스트 세트를 제공합니다. 솔루션을 직접 코딩하거나 Cyc(Curtis et al., 2005)와 같은 기존의 대규모 QA 시스템을 사용하고 싶은 유혹을 느낄 수 있습니다. 우리의 구조화된 SVM 결과(손으로 구축한 기능이 있는 계단식 NLP 시스템)가 이것이 간단하지 않다는 것을 보여주더라도 그들은 문제를 해결하는 데 성공할 수 있습니다. 그러나 이러한 접근 방식은 문제를 해결하는 방법을 배우지 않기 때문에 이것은 작업의 목적이 아닙니다. 우리의 실험은 일부 기존 기계 학습 방법이 일부 작업, 특히 유용한 확장을 도입한 메모리 네트워크에서 성공적임을 보여줍니다(섹션 A에서). 그러나 이러한 모델은 여전히 여러 작업에서 실패하고 일반적으로 현실적인 것보다 훨씬 더 강력한 형태의 감독(지원 사실 사용)을 사용합니다.

이러한 데이터 세트는 아직 해결되지 않았습니다. 향후 연구는 필요한 감독의 양을 최소화하고 새로운 작업을 해결하는 데 필요한 훈련 사례의 수를 최소화하여 인간의 작업 전달 능력에 더 가깝게 이동하는 것을 목표로 해야 합니다. 즉, 훈련 예제가 1000개 이하인 약한 감독 사례에서는 작업을 해결하는 알려진 일반적인(즉, 수동 엔지니어링되지 않은) 방법이 없습니다. 더 중요한 것은 더 어려운 작업을 개발하는 피드백 루프와 이를 해결할 수 있는 알고리즘이 우리를 유익한 연구 방향으로 이끄는 것입니다.

이러한 작업은 실제 데이터를 대체하는 것이 아니라 특히 알고리즘을 개발하고 분석할 때 이를 보완해야 합니다. 많은 보완적인 실제 데이터 세트가 있습니다(예: Hermann et al. 참조). (2015); Bordes et al. (2015); Hill et al. (2015). 즉, 방법이 20개의 작업에서 잘 작동하더라도 실제 데이터에서도 유용함을 보여야 합니다.

영향 온라인 상태가 된 이후로 bAbI 작업은 Sukhbaatar 등의 MemN2N(weakly supervised end-to-end Memory Networks)을 포함하여 몇 가지 유망한 새 알고리즘의 개발에 이미 직접적인 영향을 미쳤습니다. (2015), Kumar 등의 동적 메모리 네트워크. (2015) 및 신경 추론기(Peng et al., 2015). 이후 MemN2N은 일부 실제 작업에서 잘 수행되는 것으로 나타났습니다(Hill et al., 2015).

참조

- Bache, K. 및 Lichman, M. <http://archive.ics.uci.edu/ml>. UCI 기계 학습 저장소, 2013. URL
- Berant, Jonathan, Chou, Andrew, Frostig, Roy 및 Liang, Percy. freebase에서 시맨틱 파싱 질문-답변 쌍. EMNLP, pp. 1533–1544, 2013.
- Berant, Jonathan, Srikumar, Vivek, Chen, Pei-Chun, Huang, Brad, Manning, Christopher D, Van der Linden, Abby, Harding, Brittany 및 Clark, Peter. 독해를 위한 생물학적 과정 모델링. 프로세스 에서 EMNLP, 2014.
- Bordes, Antoine, Usunier, Nicolas, Collobert, Ronan 및 Weston, Jason. 위치 자연어를 이해하기 위해. AISTATS, 2010.
- Bordes, Antoine, Usunier, Nicolas, Chopra, Sumit 및 Weston, Jason. 대규모 간단한 질문 메모리 네트워크로 응답합니다. arXiv 사전 인쇄 arXiv:1506.02075, 2015.
- Chen, David L 및 Mooney, Raymond J. 관찰에서 자연어 내비게이션 지침을 해석하는 방법 배우기. 샌프란시스코, 캘리포니아, pp. 859–865, 2011.
- Collobert, Ronan, Weston, Jason, Bottou, L'eon, Karlen, Michael, Kavukcuoglu, Koray 및 Kuksa, Pavel. 처음부터 (거의) 자연어 처리. 기계 학습 연구 저널, 12:2493–2537, 2011.
- Curtis, Jon, Matthews, Gavin 및 Baxter, David. 질문 응답 시스템에서 cyc의 효과적인 사용에 대해. In IJCAI Workshop on Knowledge and Reasoning for Answering Questions, pp. 61–70, 2005.
- Fader, Anthony, Zettlemoyer, Luke 및 Etzioni, Oren. 개방형 질문 답변을 위한 의역 중심 학습. ACL, pp. 1608–1618, 2013.
- Fader, Anthony, Zettlemoyer, Luke 및 Etzioni, Oren. 선별 및 추출된 지식 기반에 대한 공개 질문 답변. 지식 발견 및 데이터 마이닝에 관한 제20회 ACM SIGKDD 국제 회의 진행, pp. 1156–1165. ACM, 2014.
- Graves, Alex, Wayne, Greg 및 Danihelka, Ivo. 신경 튜링 기계. arXiv 사전 인쇄 arXiv:1410.5401, 2014.
- Halevy, Alon, Norvig, Peter 및 Pereira, Fernando. 데이터의 비합리적인 효율성. 인텔리гент Systems, IEEE, 24(2):8–12, 2009.
- Hermann, Karl Moritz, Köpcke, Tomáš, Grefenstette, Edward, Espeholt, Lasse, Kay, Will, Suleyman, Mustafa 및 Blunsom, Phil. 기계에게 읽고 이해하도록 가르치는 것. 신경 정보 처리 시스템(NIPS)의 발전, 2015. URL <http://arxiv.org/abs/1506.03340>.
- Hill, Felix, Bordes, Antoine, Chopra, Sumit 및 Weston, Jason. 골디락스 원리: 명시적 기억 표상으로 아동 도서 읽기 s. arXiv 사전 인쇄 arXiv:1511.02301, 2015.
- Hochreiter, Sepp 및 Schmidhuber, Jürgen. 장단기 기억. 신경 계산, 9(8): 1735–1780, 1997.
- Kumar, Ankit, Irsoy, Ozan, Su, Jonathan, Bradbury, James, English, Robert, Pierce, Brian, Ondruska, Peter, Gulrajani, Ishaan 및 Socher, Richard. 무엇이든 물어보세요: 동적 메모리 네트워크는 자연어 처리를 위해 작동합니다. <http://arxiv.org/abs/1506.07285>, 2015.

- Levesque, Hector J, Davis, Ernest 및 Morgenstern, Leora. winograd 스키마 챌린지. AAAI 봄 심포지엄: 상식 추론의 논리적 형식화, 2011.
- 리앙, 퍼시. Lambda 종속성 기반 구성 의미 체계. arXiv 프리프린트 arXiv:1309.4408, 2013.
- Liang, Percy, Jordan, Michael I, Klein, Dan. 종속성 기반 구성 의미 학습. 전산 언어학, 39(2):389–446, 2013.
- Minsky, Marvin 및 Papert, Seymour. 퍼셉트론: 계산 기하학 소개. 그만큼 MIT Press, Cambridge, 확장판, 19:88, 1969.
- 몽포르, 닉. Twisty Little Passages: 인터랙티브 픽션에 대한 접근 방식. 미트 프레스, 2005.
- Müller, KR, Smola, Alex J, Ratsch, Gunnar, Schölkopf, Bernhard, Kohlmorgen, Jens 및 Vapnik, Vladimir. 서포트 벡터 머신으로 시계열을 예측합니다. In Artificial Neural Networks I CANN'97, pp. 999–1004. 스프링거, 1997.
- Ng, Andrew Y, Jordan, Michael I, Weiss, Yair 등. 스펙트럼 클러스터링: 분석 및 알고리즘. 신경 정보 처리 시스템의 발전, 2:849–856, 2002.
- Peng, Baolin, Lu, Zhengdong, Li, Hang 및 Wong, Kam-Fai. 신경망 기반 추론을 향하여. arXiv 사전 인쇄 arXiv:1508.05508, 2015.
- Raghuathan, Karthik, Lee, Heeyoung, Rangarajan, Sudarshan, Chambers, Nathanael, Surdeanu, Mihai, Jurafsky, Dan 및 Manning, Christopher. 공동 참조 분해능을 위한 다중 통과 체. 자연어 처리의 경험적 방법에 관한 2010년 회의 진행, pp. 492–501. 전산 언어학 협회, 2010.
- Richardson, Matthew, Burges, Christopher JC 및 Renshaw, Erin. MCTest: 챌린지 데이터 세트 텍스트의 개방형 도메인 기계 이해. EMNLP, pp. 193–203, 2013.
- Rumelhart, David E, Hinton, Geoffrey E, Williams, Ronald J. 오류 전파를 통한 내부 표현 학습. 기술 보고서, DTIC 문서, 1985.
- Soon, Wee Meng, Ng, Hwee Tou, 그리고 Lim, Daniel Chung Yong. 명사구의 상호 참조 해결에 대한 기계 학습 접근 방식입니다. 전산 언어학, 27(4):521–544, 2001.
- Sukhbaatar, Sainbayar, Szlam, Arthur, Weston, Jason 및 Fergus, Rob. 엔드투엔드 메모리 네트워크. NIPS 간행물, 2015.
- Sutskever, Ilya, Vinyals, Oriol 및 Le, Quoc VV. 신경망을 이용한 시퀀스 투 시퀀스 학습. 신경 정보 처리 시스템의 발전, pp. 3104–3112, 2014.
- UzZaman, Naushad, Llorens, Hector, Allen, James, Derczynski, Leon, Verhagen, Marc 및 Pustejovsky, James. Tempeval-3: 이벤트, 시간 표현 및 시간 관계 평가. arXiv 사전 인쇄 arXiv:1206.5333, 2012.
- Weston, Jason, Chopra, Sumit 및 Bordes, Antoine. 메모리 네트워크. CoRR, abs/1410.3916, 2014.
- 위노그라드, 테리. 자연어 이해. 인지 심리학, 3(1):1–191, 1972.
- Yao, Xuchen, Berant, Jonathan, Van Durme, 벤자민. Freebase qa: 정보 추출 또는 시맨틱 파싱? ACL 2014, 82페이지, 2014.
- Yu, Mo, Gormley, Matthew R 및 Dredze, Mark. 요인 기반 구성 임베딩 모델. 학습 의미론에 관한 NIPS 2014 워크샵, 2014.
- Zhu, Xiaojin, Ghahramani, Zoubin, Lafferty, John 등. 가우시안 필드와 하모닉 함수를 사용한 준지도 학습. ICML, 3권, pp. 912–919, 2003.

메모리 네트워크의 확장

메모리 네트워크 Weston et al. (2014)는 QA에서 잘 수행되는 것으로 나타났으며 작업에 적용할 수 있는 유망한 모델 클래스입니다. 이들은 메모리 m (m_i 로 인덱싱된 객체 배열)와 입력이 주어지면 실행되는 4개의 잠재적으로 학습 가능한 구성 요소 I , G , O 및 R 로 구성됩니다.

I : (입력 특징 맵) - 입력 문장 x 를 내부 특징 표현 $I(x)$ 로 변환합니다.

G : (일반화) - 새로운 입력이 주어지면 현재 메모리 상태 m 을 업데이트합니다: $m_i = G(m_i, I(x), m), \forall i$.

O : (출력 가능 맵) - 새로운 입력과 메모리가 주어지면 출력 o 를 계산합니다. $o = O(I(x), m)$.

R : (응답) - 마지막으로 출력 가능 o 를 디코딩하여 사용자에게 최종 텍스트 응답을 제공합니다. $r = R(o)$.

잠재적으로 구성 요소 I 는 파싱 및 엔터티 해결과 같은 표준 전처리를 사용할 수 있지만 가장 간단한 형태는 전혀 처리하지 않는 것입니다. G 의 가장 간단한 형태는 새로 들어오는 예제를 빈 메모리 슬롯에 저장하고 나머지 메모리는 그대로 두는 것입니다. 따라서 Weston et al. (2014) 사용된 실제 구현은 정확히 이 간단한 형식이며 작업의 대부분은 O 및 R 구성 요소에 있습니다. 전자는 메모리에서 읽고 추론을 수행하는 책임이 있습니다. 예를 들어 질문에 답하기 위해 관련 메모리가 무엇인지 계산하고 후자는 O 가 주어진 대답의 실제 문구를 생성하는 것입니다.

O 모듈은 주어진 x 를 지원하는 k 개의 메모리를 찾아 출력 기능을 생성합니다. 그들은 $k = 2$ 를 사용합니다. $k = 1$ 인 경우 가장 높은 점수 지원 메모리는 다음과 같이 검색됩니다.

$$o_1 = O_1(x, m) = \text{인수 최대값}_{i=1, \dots, N} sO(x, m_i) \quad (1)$$

여기서 sO 는 문장 쌍 x 와 m_i 사이의 일치 점수를 매기는 함수입니다. $k = 2$ 인 경우 이전 반복에서 찾은 첫 번째 메모리가 주어지면 두 번째 지원 메모리를 찾습니다.

$$o_2 = O_2(q, m) = \text{인수 최대}_{i=1, \dots, N} sO([x, mo_1], m_i) \quad (2)$$

여기서 후보 지원 메모리 m_i 는 이제 원래 입력과 첫 번째 지원 메모리 모두에 대해 점수가 매겨지며 대괄호는 목록을 나타냅니다. 최종 출력 o 는 모듈 R 에 입력되는 $[x, mo_1, mo_2]$ 입니다.

마지막으로 R 은 텍스트 응답 r 을 생성해야 합니다. 저자는 순환 신경망(RNN)도 고려하지만 표준 설정은 응답을 순위를 매겨 단일 단어(모델이 본 모든 단어 중에서)로 제한합니다.

$$r = R(q, w) = \text{argmax}_{w \in W} sR([x, mo_1, mo_2], w) \quad (3)$$

여기서 W 는 사전에 있는 모든 단어의 집합이고 sR 은 일치 점수를 매기는 함수입니다.

스코어링 함수 sO 및 sR 은 임베딩 모델과 동일한 형식을 갖습니다: $s(x, y) = \Phi(x) \cdot U \cdot \Phi(y)$.

(4)

여기서 U 는 $\times D$ 행렬입니다. 여기서 D 는 피쳐 수이고 n 은 임베딩 차원입니다.

Φ_x 와 Φ_y 의 역할은 원본 텍스트를 D 차원 특징 공간에 매핑하는 것입니다. 그들은 bag of words 표현을 선택하고 $D = 3|W|$ sO 의 경우, 즉, 사전의 모든 단어는 세 가지 다른 표현을 가집니다: 하나는 $\Phi_y(\cdot)$ 에 대한 것이고 두 개는 $\Phi_x(\cdot)$ 에 대한 것입니다. 다르게 모델링할 수 있습니다.

그들은 모델의 다양한 확장, 특히 쓰기 시간 모델링과 보이지 않는 단어 모델링을 고려합니다. 여기서는 우리가 사용하는 전자에 대해서만 논의합니다. 모델이 이야기에 대한 QA 작업을 수행하려면 모델에서 직접 사용할 수 없는 문장이 발화된 순서를 알아야 합니다. 따라서 SO 에 추가 작성 시간 추가 기능을 추가하여 비교 중인 다른 문장보다 오래된 문장을 나타내는 값 0 또는 1을 취하고 세 쌍의 문장과 질문 자체를 비교합니다. 교육은 질문 답변 쌍과 지원 메모리(o_1 및 o_2 선택) 모두의 감독을 사용하여 확률적 경사 하강법으로 수행됩니다. 웨스턴 등을 참조하십시오. (2014) 자세한 내용은 참조하십시오.

A.1 기존 MEMNNS의 단점

(Weston et al., 2014)에 정의된 메모리 네트워크 모델은 작업을 시도할 수 있는 가능한 기술 중 하나이지만 실패할 가능성이 있는 몇 가지 작업이 있습니다.

- 단어 모음으로 문장을 모델링하므로 2-인수(작업 4) 및 3-인수(작업 5) 관계 문제.
- 두 개의 최대 연산($k=2$)만 수행하므로 관련 질문을 처리할 수 없습니다. 작업 3 및 7과 같은 두 개 이상의 지원 사실.
- RNN이 R 모듈에 사용되지 않는 한 eq를 사용하는 표준 설정에서 다중 답변을 제공할 수 없습니다. (삼). 목록(8) 및 경로 찾기 (19) 작업에 필요합니다.

따라서 다음 섹션에서 모델에 대한 개선 사항을 제안합니다.

A.2 메모리 네트워크 개선

A.2.1 적응형 메모리 (및 응답)

우리는 문는 질문에 따라 자동으로 조정되는 가변적인 수의 지원 사실을 고려합니다. 이를 위해 특별한 사실 $m \emptyset$ 점수를 매기는 것을 고려합니다. 그러면 지원 메모리의 계산은 다음과 같이 됩니다.

```


$$\begin{aligned}
 & \text{나는} = 1 \\
 & \text{oi} = O(x, m) \text{ 동} \\
 & \text{안 } \text{oi} = m \emptyset \text{ do } i \leftarrow \\
 & \quad i + 1 \text{ oi} = O([x, \\
 & \quad \text{mo1}, \dots, \text{moi} - 1], m) \text{ end while}
 \end{aligned}$$


```

즉, $m \emptyset$ 가 우리가 멈추는 지점에서 예측될 때까지 이전에 발견된 사실에 대한 각 단계에서 조건을 지정하면서 지원 사실 i 를 계속 예측합니다. $m \emptyset$ 에는 자체 고유한 임베딩 벡터가 있으며 이 벡터도 학습됩니다. 실제로 우리는 계산이 멈추지 않는 실패 사례를 피하기 위해 여전히 실험에서 최대 루프 수를 부과합니다(우리 실험에서는 10의 제한을 사용합니다).

Multiple Answers 여러 단어를 출력하기 위해 응답 모듈에도 유사한 트릭을 사용합니다. 즉, 우리는 사전에 특수 단어 $w \emptyset$ 를 추가하고 이전 단어에 대한 조건으로 각 반복 i 에서 단어 w_i 를 예측합니다. 즉, $w_i = R([x, \text{mo1}, \dots, m|i], \dots, w_{i-1}, w)$, $w \emptyset$ 를 예측할 때까지.

A.2.2 비선형 문장 모델링

단어 모음을 넘어서는 문장을 모델링하는 방법에는 여러 가지가 있으며 여기서는 세 가지 변형을 살펴봅니다. 가장 간단한 것은 bag-of-N-grams이며, 우리는 bag에서 $N = 1, 2, 3$ 을 고려합니다. 이러한 방법의 주요 단점은 사전이 N 과 함께 빠르게 증가한다는 것입니다. 따라서 우리는 다중 선형 맵이라고 하는 대안적인 신경망 접근 방식을 고려합니다. 문장의 각 단어는 $p(i, l) = (i \text{Ps}z)/l$ 을 사용하여 $\text{Ps}z$ 위치 중 하나로 비닝됩니다. 여기서 i 는 길이가 l 인 문장에서 단어의 위치이며 각 위치에 대해 다음을 사용합니다. $\times n$ 행렬 $Pp(i, l)$. 그런 다음 일치하는 점수를 다음과 같이 모델링합니다.

$$s(q, d) = E(q) \cdot E(d); E(x) = \tanh\left(\sum_{i=1, \dots, l} PP(i, l) \Phi(x_i) \cdot U\right) \quad (5)$$

여기서 우리는 위치에 따라 각 단어에 대해 선형 맵을 적용한 다음 매핑 합계에 대한 $\tanh$ 비선형성을 적용합니다. 이는 위치보다 태그를 고려하는 (Yu et al., 2014) 모델과 관련이 있습니다. 이 방법의 결과는 공간 제한으로 인해 본문에 표시되지 않았지만 N-그램과 유사하게 잘 수행되며 N-그램으로 인해 사전이 너무 커지는 실제 사례에 유용할 수 있습니다. 적응형 메모리(AM) + 다중 선형이 포함된 표 3 MemNN과 비교하면 AM+NG+NL(즉, N-gram을 대신 사용)이 포함된 MemNN과 동일한 93의 평균 성능을 얻습니다.

마지막으로 단어 위치를 전혀 모델링하지 않는 비선형 맵의 성능을 평가하기 위해 다음과 같은 비선형 임베딩도 고려합니다.

$$E(x) = \tanh(W \tanh(\Phi x(x) - U)). \quad (6) \text{ 여기서 } W \text{는 } \times n \text{ 행렬입니다.}$$

이것은 고전적인 2계층 신경망과 유사하지만 $s(q, d)$ 의 양변 q 와 d 에 적용됩니다. 우리는 또한 이 비선형성이 뒤따르는 bag-of-N-gram의 간단한 조합을 고려합니다.

B 외부 리소스를 사용하는 기준선

우리는 또한 구조화된 SVM을 사용하여 고전적인 캐스케이드 NLP 시스템 기준선을 구축했습니다. 이 기준선은 비용이 많이 드는 대량의 레이블이 지정된 데이터에 대해 자체적으로 훈련되는 상호 참조 해상도 및 의미론적 역할 레이블 지정 전처리 단계를 통합합니다. 우리는 먼저 이야기에 대해 Stanford 공동 참조 시스템(Raghunathan et al., 2010)을 실행하고 각 언급은 엔티티 클래스의 첫 번째 언급으로 대체됩니다. 둘째, SENNA 의미론적 역할 라벨링 시스템(SRL)(Collobert et al., 2011)을 실행하고 각 동사에 대한 인수 집합을 수집합니다. 그런 다음 지원 사실을 찾기 위한 순위 작업을 정의합니다(강력한 감독을 사용하여 훈련됨).

$$o_1, o_2, o_3 = \text{인수 최대값 } o \in O \quad SO(x, fo_1, fo_2, fo_3, \Theta)$$

질문 x 가 주어졌을 때 우리는 이야기의 사실 집합 f 에서 인덱스 o_i 가 있는 최대 3개의 지원 사실을 찾고 (또한 3개 미만의 경우 "빈 사실"을 선택하는 것도 고려합니다) SO 는 선형 점수 함수입니다. 매개변수 Θ . argmax 를 계산하려면 탐욕적인 MemNN 방법과 달리 철저한 검색을 수행해야 합니다. 따라서 확장성을 위해 우리는 사실이 하나의 공통 비결정자 단어를 서로 일치 또는 x 와 공유하도록 요구하여 가능한 일치 집합을 정리합니다. SO 는 일련의 지표 기능으로 구성됩니다. 단순성을 위해 각 기능은 문장 쌍만 본다면. 즉, $SO(x, fo_1, fo_2, fo_3; \Theta) = \Theta \quad (g(x, fo_1), g(x, fo_2), g(x, fo_3), g(fo_1, fo_2), g(fo_2, fo_3), g(fo_1, fo_3))$. 기능 함수 g 는 $g(fo_1, fo_2)$ 에 대해 다음과 같은 기능 유형으로 구성됩니다. (1) 단어 쌍: fo_1 및 fo_2 의 각 단어 쌍에 대한 하나의 표시기 변수입니다. (2) 쌍 거리: 문장 사이의 거리를 나타내는 지표, 즉 $o_1 - o_2$. (3) 쌍 순서: 문장의 순서를 나타내는 지표, 즉 $o_1 > o_2$. (4)

SRL 동사 쌍: fo_1 및 fo_2 의 각 SRL 동사 쌍에 대한 지시자 변수입니다. (5) SRL Verb-Arg Pair: fo_1, fo_2 및 해당 동사의 각 SRL 인수 쌍에 대한 지시자 변수.

지원 사실을 찾은 후 응답 단계에 대해 유사한 구조화된 SVM을 구축하고 해당 목표에 맞게 조정된 기능도 사용합니다. 단어 $-x$ 의 각 단어에 대한 표시기, 단어 쌍 $-x$ 의 각 단어 쌍에 대한 표시기 및 지원 사실 이전과 유사한 SRL Verb 및 SRL Verb-Arg 쌍 기능.

결과는 표 3에 나와 있습니다. 구조화된 SVM은 외부 리소스에 대한 액세스 권한이 있음에도 불구하고 전반적으로 MemNN보다 성능이 좋지 않으며 여전히 9개의 작업에서 실패합니다. 수작업으로 만든 기능 결합이 원래 MemNN이 하지 못하는 필요한 비선형성을 캡처하는 작업 6, 9 및 10에서 잘 수행됩니다. 그러나 3개(때로는 2개)의 지원 사실(예: 작업 3, 16 및 2)이 필요한 작업에서는 아마도 너무 많은 가능성에 순위를 매기면 더 많은 실수가 발생하기 때문에 훨씬 더 나쁜 것으로 보입니다. 그러나 탐욕스럽지 않은 검색은 검색이 매우 중요한 경로 찾기(작업 19)와 같은 다른 작업에 도움이 되는 것 같습니다.