비디오 행동들의 설명 가능한 예측을 위한 효율적인 구문 분석

이준석O 공다영 정든솔 곽수하 조민수
포항공과대학교
{jameslee, dayoung.gong, deunsol.jung, suha.kwak, mscho}@postech.ac.kr

Efficient grammar parsing for explainable prediction of action sequences in videos

Joonseok LeeO Dayoung Gong Deunsol Jung Suha Kwak Minsu Cho
POSTECH

요 약

동영상이나 음성과 같은 데이터의 분석은 여러 단계적 의미 구조를 이해하는 것이 중요하다. 문맥 자유 문법(Context-Free Grammar)은 이러한 속성을 표현할 수 있으며 분석 트리를 통해 의사결정 과정을 가시화 할 수 있다는 장점이 있다. 그러나 기존의 구문 분석기는 연속적인 데이터를 입력으로 사용할 수 없거나, 특정 형태의 문법을 분석할 수 없다. 본 논문에서는 이러한 문제를 해결하고자 너비 우선 탐색과 가지치기 방식을 도입한 새로운 확률 기반 구문 분석기를 제안한다.

1. 서 론

동영상에서의 활동 (activity)은 하나 이상의 행동 (action)으로 구성되며, 활동의 목표, 개별 행동의 특성, 물리적 환경 등에 의해 구조화된다. 따라서 행동 분석에서 활동의 단계적 의미 구조를 이해하는 것은 필수적인 요소이다. 문법은 언어의 계층 구조를 명시적으로 자연스럽게 표현하는 방법이며, 이는 활동의 구조를 나타내는 것에도 동일하게 적용할 수 있다. 이에 더해, 문법은 구문 분석을 통해 분석 트리를 생성할 수 있으며, 생성된 트리는 모델이 어떤 생성 규칙을 통해 행동 사이의 관계를 규정하는지 파악할 수 있다는 장점이 있다. 그러나 전통적인 문법 구문 분석기 (parser)는 언어에 맞추어 설계된 만큼, 동영상과 같은 연속적인 데이터를 처리할 수 없다는 단점이 존재했다. 이를 해결하고자 일반화된 형태의 Earley 분석기 (Generalized Earely Parser, GEP) [1]이 제안되었으나, 둘 이상의 생성규칙을 가지는 변수 (variable)를 포함하는 문법을 분석할 때 분석 트리의 가지가 급증하여 합리적인 시간 내에 분석을 마치지 못하는 문제가 발생했다. 이러한 문제를 해결하기 위해, 우리는 본 논문에서 Earley 분석기 [2]에 너비 우선 탐색과 가지치기 기법을 적용한 효과적인 구문 분석기를 제안한다. 제안하는 분석기는 훈련 데이터로부터 추출한 문맥 자유 문법과 프레임 수준의 확률 데이터를 기반으로 가장 가능성 있는 행동들의 시퀀스를 출력한다.

본 논문의 구성은 다음과 같다. 2장에서는 동영상 분석 연구 중 하나인 시간적 행동 분할에 대해서 설명하고, 3장에서는 제안하는 너비 우선 Earley 분석기에 대해 서술한다. 4장에서는 제안한 구문 분석기를 적용한 결과를 소개하고 마지막으로 5장에서는 결론을 맺는다.

2. 시간적 행동 분할 (Temporal action segmentation)

시간적 행동 분할이란 T 프레임의 동영상 $f=[f_1,f_2,\cdots,f_T]$와 행동 집합 A가 주어졌을 때, 가장 적절한 행동 시퀀스 $a=[a_1,a_2,\cdots,a_N]$와 각 행동에 해당하는 프레임의 길이 $l=[l_1,l_2,\cdots,l_N]$로 이루어진 쌍을 출력하는 것을 목표로 한다. 이때 a의 모든 원소는 A에 속하며, $1\le i\le N-1$인 i에 대해 $a_i\neq a_{i+1}$이고, $\sum_{i=1}^{N} l_i = T$를 만족한다.

시간적 행동 분할은 동영상 내의 행동들을 배열하는 과제인 만큼, 행동 간의 인과관계나 대체관계 등의 단계적 의미 구조를 이해하는 것이 중요하다. 이런 속성을 학습하기 위해 기존의 연구에서는 그래프를 사용하거나 [3], 행동 클래스 단위의 표현 (Representation)을 추출하여 결과를 보정하는 형식 [4]을 취했다. 다만 이러한 구조들은 의사결정 과정이나 입력으로 사용되는 요소들이 사람이 이해하기 어렵다.

본 논문에서 제안하는 구문 분석기는 훈련 데이터 집합으로부터 행동 클래스를 단말 (terminal)로 하는 문맥 자유 문법을 추출하고, 사전 학습된 시간적 행동 분할 모델의 출력을 이용하였다. 분석기는 문법과 모델의 출력을 입력으로 받아 행동 시퀀스 a^*를 생성하며, 이후 동적 프로그래밍 기반의 Viterbi decoding [5]을 통해 분석기의 결과에 가장 적합한 길이 l^*을 생성한다.

3. 너비 우선 Earley 분석

우리는 GEP [1]에서 제안한 구문 분석 확률에 너비 우선 탐색과 가지치기 기법을 적용한 효과적인 구문 분석기를 소개한다. 제안된 분석기는 Earley 분석기에서의 각 상태 (State)마다 분석 트리에서의 깊이를 식별하고, 그 깊이를 기준으로 우선순위를 정렬한다. 분석 과정 중 탐색한 상태는 기록되어 분석 트리를 구성하며, 기록된 상

태들의 시퀀스 확률과 접두사 확률을 통해 정량적으로 비교가 가능하다. 또한, 큐 (Queue)에 크기 제한을 두고 접두사 확률이 낮은 상태를 제거함으로써 분석 트리의 탐색 공간을 줄이는 가지치기 방식을 도입하였다.

3.1. 구문 분석 확률

구문 분석 확률은 사전 학습된 시간적 행동 분할 모델의 출력 $Y \in \mathbb{R}^{T \times |A|}$을 통해 계산하며, 두 가지로 나누어진다. 하나는 시퀀스 확률 (sequence probability)로, 특정 프레임 t까지의 구문 분석 결과가 a일 확률을 의미한다. 시퀀스 확률은 a의 마지막 행동 클래스가 z라고 가정할 때, 다음과 같이 정의된다.

$$p(f_{1:t} \rightarrow a) = Y_{t,z}\{p(f_{1:t-1} \rightarrow a) + p(f_{1:t-1} \rightarrow a_{1:|a|-1})\} \quad (1)$$

다른 하나는 행동 시퀀스 a가 주어진 동영상의 구문 분석 결과의 일부일 경우의 확률, 접두사 확률 (prefix probability)이다.

$$p(a\ldots) = p(f_1 \rightarrow a) + \sum_{t=2}^{T} Y_{t,z}\, p(f_{1:t-1} \rightarrow a_{1:|a|-1}) \quad (2)$$

3.2. 구문 분석 과정

구문 분석 과정은 Earley 분석 [2]과 유사한 구조로 구성되어 있으며, 예측 (Prediction), 스캔 (Scanning), 완성 (Completion)의 세 가지 작업으로 이루어져 있다. 각 작업은 분석의 상태를 생성하고 수정하며, 작업 이후 생성되거나 수정된 상태는 큐에 저장되어 분석 트리 내에서의 깊이에 따라 처리 우선순위가 정해진다. 각 상태는 처리 중인 생성규칙 r, 현재 상태를 생성한 부모 상태, 현재까지 분석 완료된 행동 시퀀스 a와 접두사 확률 $p(a\ldots)$로 이루어져 있다.

- 예측 (Prediction): $S(m,n,d)$에 속하는 상태 $(A \rightarrow \alpha \bullet B\beta, S(i,j,k), a, p(a\ldots))$에 대해, 좌변이 B인 모든 생성 규칙을 찾고, $(B \rightarrow \bullet\, \Gamma, S(m,n,d), a, p(a\ldots))$의 형태로 $S(m,n,d+1)$에 추가한다.
- 스캔 (Scanning): $S(m,n,d)$에 속하는 상태 $(A \rightarrow \alpha \bullet x\beta, S(i,j,k), a, p(a\ldots))$에 대해, $p((a+x)\ldots)$를 계산한다. 그 이후 $S(m+1, n', d)$에 $(A \rightarrow \alpha x \bullet \beta, S(i,j,k), a+x, p((a+x)\ldots))$를 추가한다. 이때 n'은 $S(m+1)$의 크기가 된다.
- 완성 (Completion): $S(m,n,d)$에 속하는 상태 $(A \rightarrow \Gamma \bullet, S(i,j,k), a, p(a\ldots))$에 대해, $(B \rightarrow \alpha \bullet A\beta, S(i',j',k'), a', p(a'\ldots))$의 형태를 가지고 있는 상태를 $S(i,j,k)$에서 찾고, $S(m,n,d-1)$에 $(B \rightarrow \alpha A \bullet \beta, S(i',j',k'), a', p(a'\ldots))$를 추가한다.

위의 과정에서 α, β, Γ는 변수와 행동 클래스가 포함된 시퀀스를 나타내며, A, B는 변수, x는 행동 클래스, 중앙점 $\bullet$은 현재 분석기의 분석 중인 위치를 나타낸다.

예측, 스캔, 완성 과정이 한 차례 끝나면, 상태가 저장된 큐를 각 상태의 접두사 확률에 대해 내림차순으로 정렬한다. 그리고 주어진 큐 크기 제한에 따라 낮은 확률을 가진 상태들을 제거하는 가지치기 과정을 거친다. 이후 큐에 저장된 깊이가 가장 작은 상태에 대해 위의 과정을 반복한다. 구문 분석은 모든 상태가 완료되어 큐가 비었거나 지금까지 분석이 완료된 행동 시퀀스 중 하나가 큐에 남아있는 다른 모든 상태의 접두사 확률보다 시퀀스 확률이 높아 더 이상 탐색할 필요가 없는 경우 종료되고, 시퀀스 확률이 가장 높은 시퀀스를 출력한다.

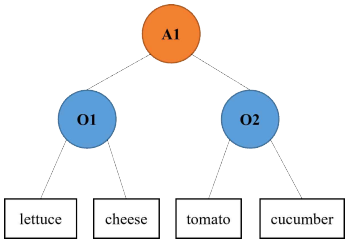

그림 1 추출된 문법 예시

Frame	P("lettuce")	P("cheese")	P("tomato")	P("cucumber")
1	0.7	0.1	0.1	0.1
2	0.7	0.1	0.1	0.1
3	0.1	0.1	0.7	0.1
4	0.1	0.1	0.3	0.5

그림 2 각 프레임 별 사전 학습된 분할 모델 출력

Frame	lettuce	cheese	lettuce - tomato	lettuce - cucumber
1	0.7	0.1	0.1	0.1
2	0.49	0.01	0.07	0.07
3	0.049	0.001	0.343	0.049
4	0.0049	0.0001	0.1029	0.0245

그림 3 각 프레임 별 시퀀스 확률

(m, n, d)	Production rule	Parsed sequence	Operation
(0, 0, 0)	$S \rightarrow \cdot\, A1$	ε	
(0, 0, 1)	$A1 \rightarrow \cdot\, O1\, O2$	ε	PRED from (0, 0, 0)
(0, 0, 2)	$O1 \rightarrow \cdot$ "lettuce"	ε	PRED from (0, 0, 1)
(0, 0, 2)	$O1 \rightarrow \cdot$ "cheese"	ε	PRED from (0, 0, 1)
(1, 0, 2)	$O1 \rightarrow$ "lettuce" $\cdot$	lettuce	SCAN from (0, 0, 2)
(1, 1, 2)	$O1 \rightarrow$ "cheese" $\cdot$	cheese	SCAN from (0, 0, 2)
(1, 0, 1)	$A1 \rightarrow O1 \cdot O2$	lettuce	COMP from (1, 0, 2)
(1, 0, 2)	$O2 \rightarrow \cdot$ "tomato"	lettuce	PRED from (1, 0, 1)
(1, 0, 2)	$O2 \rightarrow \cdot$ "cucumber"	lettuce	PRED from (1, 0, 1)
(2, 0, 2)	$O2 \rightarrow$ "tomato" $\cdot$	lettuce – tomato	SCAN from (1, 0, 2)
(2, 1, 2)	$O2 \rightarrow$ "cucumber" $\cdot$	lettuce - cucumber	SCAN from (1, 0, 2)
(2, 0, 1)	$A1 \rightarrow O1\, O2 \cdot$	lettuce – tomato	COMP from (2, 0, 2)
(2, 0, 0)	$S \rightarrow A1 \cdot$	lettuce – tomato	COMP from (2, 0, 1)

그림 4 분석 과정 예제

분석 과정의 예제는 그림 1-4와 같다. 입력으로 사용하는 문법은 그림 1과 같으며, 해당 그림은 추출된 문법을 AND-OR 그래프로 표시한 것으로, AND는 하위 노드를 모두 방문하는 노드를, OR 노드는 하위 노드 중 하나만을 방문하는 노드를 의미한다. 그림 2는 입력으로 사용한 사전 학습된 시간적 분할 모델의 각 프레임 별 행동 클래스의 확률을 나타낸다. 그림 3은 시퀀스에 따른 프

레임 별 시퀀스 확률을 보여준다. 그림 4는 분석 중 방문하는 상태와 속하는 생성규칙 및 분석한 시퀀스를 저장하고, 해당 상태가 어디서 파생되었는지를 나타낸다.

4. 실험 결과

4.1. 실험 설정

실험을 위해 50Salads 데이터 집합 [6]을 활용하였다. 50Salads는 25명의 사람이 등장하여 샐러드를 만드는 동영상 50개로 구성되어 있으며, 데이터 집합은 17개의 행동으로 구분되어 있다. 사전 학습된 시간적 행동 분할 신경망으로는 ASFormer [7]를 사용하였다. 평가 방식은 기존의 시간적 행동 분할 연구들에서 사용한 추출한 행동 편집 거리와 F1 점수, 그리고 프레임 수준 정확도로 구성된다. 편집 거리는 레벤슈타인 거리 [8]에 기반하며, 높을수록 실제 값과 비교했을 때 변경이 적게 필요하다는 것을 의미한다. 기존에 제안된 GEP [1]의 경우 큐의 크기를 제한하지 않으면 특정 활동에 대해 구문 분석이 불가능했기에, 동일하게 큐 크기를 제한한 상태로 진행하였다.

분석기	큐 크기	edit	F1@10	F1@25	F1@50	acc.
-	-	75.0	76.0	70.6	57.4	73.5
GEP [1]	10	73.3	81.1	79.1	72.9	84.0
	20	72.1	80.5	79.1	72.3	83.9
Ours	10	78.3	84.9	83.2	76.9	84.9
	20	**79.9**	**85.4**	**83.8**	**77.4**	**85.3**

표 1 큐 크기와 구문 분석기에 따른 실험 결과

4.2. 결과 및 분석

실험 결과는 표 1과 같다. 표 1의 첫 번째 행은 분석기를 사용하지 않은 사전 학습된 모델의 성능이다. 실험 결과에서 볼 수 있듯이, 본 논문에서 제안한 구문 분석 방식이 기존에 제안된 GEP [1]에 비해 더 좋은 성능을 달성했다. GEP의 경우 상태를 탐색하는 과정 중 접두사 확률이 높은 상태를 먼저 처리하는데, 이는 깊이 우선 탐색 방식처럼 작용하여 분석 트리에서 특정 가지에 빠져 활동의 전체적인 맥락을 보는 것에서 불리하게 작용한다. 특히 큐의 크기가 제한되는 경우, 탐색 공간이 특정 행동 클래스를 분석 결과의 일부로 포함하도록 제한되어 다른 클래스를 선택하는 경우를 탐색하기 힘들게 된다. 반면 본 논문에서 제안한 방식의 경우, 깊이가 작은 노드를 우선 방문하기에 분석 트리의 전체적인 구조를 먼저 탐색할 수 있다.

5. 결론

본 논문에서는 연속적인 데이터를 활용하는 구문 분석기를 제안하고, 이를 행동 분석 과제에 적용하여 그 성능을 평가하였다. 기존의 제안된 구문 분석기보다 더 넓은 범주의 문법을 분석 가능토록 하였으며, 해당 분석기를 사용하였을 때 성능이 향상되는 것을 확인하였다. 또한 분석 트리를 통해 추론과정에서의 의사결정 기준을 파악할 수 있게끔 하였다.

향후 연구로 시간적 행동 분할 외의 행동 분석 연구에 적용할 계획이며, 문맥 자유 문법 외의 다양한 형태의 문법에 적용시키고 그 효과를 평가할 예정이다.

Acknowledgement

이 논문은 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원 (No.2022-0-00959, 의사결정 지원을 위한 퓨샷 학습 기반 시각 및 언어에 대한 인과관계 추론 기술개발, 50%) (No.2022-0-00264, 지식기반 심층논리 신경망을 활용한 통합적 비디오 해석과 생성 연구,50%) (No.2019-0-01906, POSTECH 인공지능대학원, 10%)의 지원을 받아 수행된 연구임.

참고문헌

[1] S. Qi, B. Jia, S. Huang, P. Wei, S. C. Zhu, A generalized earley parser for human activity parsing and prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43.8: 2538-2554, 2020.

[2] J. Earley, An efficient context-free parsing algorithm. Communications of the ACM, 13.2: 94-102, 1970.

[3] Y. Huang, Y. Sugano, and Y. Sato. Improving action segmentation via graph-based temporal reasoning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 14024-14034, 2020.

[4] H. Ahn and D. Lee. Refining action segmentation with hierarchical video representations. IEEE International Conference on Computer Vision (ICCV), pages 16302-16310, 2021.

[5] A. Richard, H. Kuehne, A. Iqbal, and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7386-7395, 2018.

[6] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. 2013 ACM international joint conference on Pervasive and ubiquitous computing, pages 729-738, 2013

[7] F. Yi, H. Wen, and T. Jiang. Asformer: Transformer for action segmentation. British Machine Vision Conference (BMVC), 2021.

[8] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady, volume 10, pages 707-710. Soviet Union, 1966.