**Relation Extraction with Self-determined Graph Convolutional Networks**

Abstract

Relation Extraction = text에서 semantic relationship 얻는 방법

Graph를 만들기 위해 linguistic tools 이용!(entities가 나타나며, GCN이 graph를 encode 하는 그런 방법 = tools)

Tools => not end-to-end process => Self-determined Graph Convolutional Network 제안.(self-attention mechanism을 이용)

Self-determined graph 만들고 => GCN encoding / test on TACRED dataset => 최고 성능 => (SGCN > GCN), traditional GCN은 parsing tool을 쓴다(그래프 만들기 위해).

1. Introduction

“Steve Jobs and Wozniak co-founded Apple in 1976” => founded\_by relation!

RE는 information extraction의 subtask => NLP/IR(information retrieval) task<QA, KG completion, Semantic Search>에서 중요한 역할을 한다.

이전 RE는 feature engineering! = linguistic, lexical tools에 의존(정보 얻기 위해) / 또한 sparse feature representation에 의해 성능 저하.

딥러닝 기반 RE 모델은 sequence-based 과 graph-based 두 가지로 나뉜다. (임베딩은 pre-trained model 써서 해결하고)

Sequence-based => embed text(using CNN or RNN) 위해 word sequence가 사용됨.

Graph-based => text -> graph 변환 후(dependency parser 나 linguistic tool 써서) GNN과 함께 processed 된다. (neighborhood 와 feature info encode해줌) -> 그리고 encoded된 feature가 RE에 쓰임.

BiLSTM이나 GCN 등이 graph encode에 사용된다. = g-based가 s-based보다 text의 먼 거리도 잘 반영? = long-distance dependency!! (Zhang 2018)

하지만 아직 text->graph를 tools을 통해 작업한다. = computationally expensive = not end-to-end training / S-based는 tool에 의존하진 않지만 long-distance dependencies가 떨어짐 = long text에 약하다.

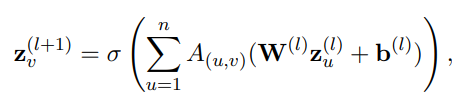
SCGN 제안 => Transformer의 self-attention mechanism(Vaswani 2017)을 이용해 graph를 추론 => self -determined graph가 만들어짐. => GCN을 통해 encoding됨.

1. G-based model to encode text without tools
2. Effectiveness on RE task
3. ablation study 는 모델이나 알고리즘의 "feature" 들을 제거하면서 그게 성능에 어떤 영향을 줄지 연구하는 것이다.

2. Graph Convolutional Network

GCN(2017, Kipf가 발표한 듯) = CNN의 확장 버전(graph에서 neighborhood info를 encode!)

Graph는 다음과 같이 표현 G = (V, A, X) <A는 sparse한 인접 행렬, X는 node embeddings, V Vertex Set>



위의 식: Each GCN layer는 node embedding from previous layer 가지고 인접 행렬이 입력, 출력은 node representations를 update!

N = 노드 개수 인듯. 모든 n->v에 대해 시그마 표현은 생략인가?

각 노드 => 차원이 d? => 거기에 W를 곱해주나? => GCN 구조랑 차원 살펴보기!!!

3. Self-determined Graph Convolution

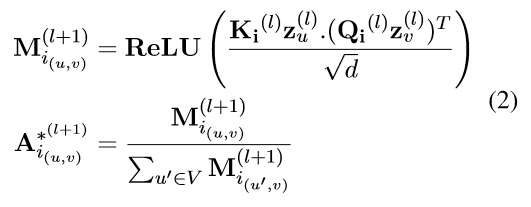
Pre-built graphs require tools!

Good results but computationally expensive => not end-to-end, low-resources languages에 적용 불가 => multiple weighted graphs를 self-determine! & 각 그래프에 separate GCN?

Word를 node로! & multiple 인접 행렬 배움( 모든 layer에 있는 )

기존의 GCN과는 다르게, A-i들은 binary가 아니라 mean-normalized real numbers(graph에서 더 강렬한 connection!)

Vaswani 2017과는 다르게 dot-product attention이 아니고

이다.

K와 Q는 d\*d dimension => layer에서 얻고자 하는 것은 최종 A-i를 말한다.

또한, 모든 노드간에 dense한 mutual connection이 없기 때문에 => ReLU가 적당.

즉, 각 레이어에서 A-i가 다 다르다. = layer마다 다른 connection

4. RE with SGCNs

기본 정보 – text T = w1, w2 … / target (관계의 대상 2개) = e1, e2  
input = a triplet(우리가 아는 그 트리플과는 좀 다름) = (e1, e2, T) / output = (e1, e2)에 대한 relation ? (관계가 없을 땐 no relation이라 나옴) / 추론할 relation의 종류는 미리 정의됨(그 pool에서 선택하는 것인가..?)

\* text -> vector sequence로 바꾼다. ( pre-trained word embedding 이용)

\* context info를 얻기 위해 biLSTM encode ! => 이렇게 나온 것이 graph 표현에 사용할 것

\* SGCN은 long-distance context에 강함, (determine the weighted connections/ propagate neighborhood information into nodes)

\* a layer aggregation( Xu 2018) / SGCN의 layer output에 대해.( = BiLSTM output) = concatenated!

\* relation classification(=relation extraction인데, 왜 classification이냐면 pool에서 선택하니까 그런듯)을 위해 (Zhang 2018) 인용 => sentence representation 과 both target entity representations를 concat한 softmax 작업을 feed-forward에 적용!

\* sentence representation 과 both target entity representations은 전체 sequence의 max-pooling, final representation에서 entitiy들의 position에 average pooling!

\* Zhang 2018에서는 C-SGCN??

음..? 이게 text -> graph 작업을 해주는건지 graph -> ~ 로 encode 작업을 해주는건지 헷갈린다.