**Parasitic Neural Network for Zero-Shot Relation Extraction**

Abstract

기존 RE => classes pool 에서 identify 가능 / 데이터셋에 없던 unseen relation type은 not recognize! => 목표: zero-shot RE(unseen relation 추론 가능, unseen type의 label name은 필요하다.) => PNN(Parastic Neural Net) 제안 = 자동으로 annotation & training 하기 위해 seen type에 기생하며 unseen type이 정의 된다.

Shared semantic space에서 text samples의 representations feature와 unseen types의 distributions 사이의 mapping을 학습 = 중요!!!

Manual annotation or 추가적 리소스가 없음에도 성능 향상(unseen RE에서)

1. Introduction

Unstructured text는 그냥 생 문장을 말하는 것인가? 어쨌든 여기서 relation facts를 determine! (Knowledge Base 확장, downstream app에 영향)

One/few – shot (이전 방법) => RE에 효과 덜함. / 세상엔 많은 relation type! => but 이 방법들은 제한된 relation만 classification(taxonomy, type is seen & 데이터셋에 일정 숫자 있어야 가능) => unseen relation 추론 불가 => 그렇다고 예상되는 카테고리(type) 데이터셋 모으기도 힘듬.

Zero-Shot RE(ZRE) 제안 = 기존 seen(labeled) type을 학습 후 new relation identify 하는 원리이자 한계이자 장점

* Annotation cost 저하 & app efficiency 향상 = 학습 방법 중 하나의 패러다임
* But 아직 발전 x, 덜 관심
* 또한 다른 task에 transfer 가능! (reading comprehension, textual entailment(TE))

이전 ZRE 연구들 = Indirect-trick 이라고도 한다. = much unnatural descriptive info 가 필요(relation type의 이해 위해) = 많은 labeled dataset 필요?

Direct-trick = relation type pre-describe 필요 없다 = 그저 어떤 것을 추론했을 때 그 이름만 지어주면 된다.

중요!!! 부분에서 seen에 overfit 되는 것을 막고 unseen에 adapt 위해

* Unseen relation type distribution을 이해 (in shared space) = PNN
* Direct-trick => unseen RE(ZRE)
* No manual annotation & external knowledge
* Automatically learn distribution of unseen types in shared semantic space(결국 relation types의 association을 이용)
* 거의 모든 indirect-trick 보다는 뛰어나고, direct-trick 몇 개 보다는 뛰어나다?

2. Related Work

Zero-shot 학습 => 대부분 CV => Entity Typing, Event Extraction, Knowledge Graph Completion 은 최근에

기존 unsupervised 방법 => manual features, patterns or corpus-level resource의 결합 기반 = 비효율, 사람 노력 많이 듬.

그 이후로 다른 task에 transfer 하는 방법으로 많이 개발 / Levy => relation types를 다양한 파라미터의 natural language questions로! => 그리고 reading comprehension 모델을 사용(relation facts를 얻으며 questions 처리)

Text -> 전제 / relation description -> 가설로 여기면서 TE 모델의 가설의 신뢰성을 결정 짓는데 transfer 했다.

But formulate 하는데 비싸고, indirect한 transfer의 capability가 에러 발생시 RE에도 영향이 있다.

Text instance와 relation type prototypes 사이으 ㅣmapping을 하는 all-inclusive manner를 이용(direct-trick) / scarce type info -> parasitic learning !

3. Methodology

3.1 Parasitism Thought

 = seen relation types의 집합(denotation, 이름)

 = unseen relation types의 집합 

는 해당 집합의 text들! /  = labeled datasets /  = test instances ->  = predict 해야함.

Type => vector화 되면 type prototypes =  = 최종 출력 / 기존 dataset x는 로 임베딩된다. => ZRE는  이다. (test instance를 predict할 수 있는)

같은 relation의 instance => 같은 prototype으로 cluster / 다른 prototype과는 멀다. (하지만 유사한 type일수록 가까이 있다)

 = between feature representation  and type prototype  / 이 거리가 짧을수록 -> 매핑이 잘되었다.

전제 3개 존재

* 학습 데이터에서 relation type R1과 상응된 x는 거리가 0
* 또한 R2와 R1과의 거리가 R3와 R1과의 거리보다 짧다면 R2와 x와의 거리가 R3와 x와의 거리보다 짧다.

=> 숙주의 type의 instance를 고려해 기생의 type의 annotation 가능. = “Parasitism” 원리

**아래는 Algorithm 1의 내용**

1~8 lines = data creation

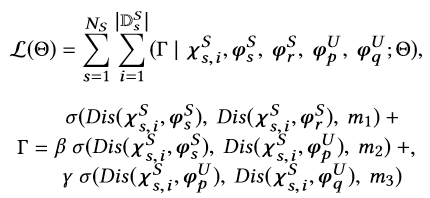
**Joint energy function**

Sub-network => semantic distance metric 생산! / metric들은 서로 상호작용, joint together!

이렇게 joint / ex) 

각 {} = trunk => branch의 semantic distance 비교

Triplet loss를 이용하며(active function은 branch1이 branch2보다 적어도 margin m보다는 작기를 원함) => ReLU function!



**3.2 Network Architecture**

4 Sub-networks => dinstinct inputs / joint energy function에 의해 joined!

Parameters of 4 sub- net => 나중에 tied! = same metric을 계산

…………………… 이해 안감……………

Zero-shot: 승용차 or not 승용차에 대해서만 학습 => 트럭이라는 클래스는 뙇하고 못 표현함 => 정확히는 그냥 승용차라고 예측(승용차에 가까운 분포 중 하나) => 의 문제를 해결

One-shot: 한 장의 데이터만으로 학습을 시키는 것

Few-shot: 소량의 데이터 만으로도 놀라운 학습 능력을 갖출 수 없을까? (인간은 소량의 데이터만 봐도 다른 것들 잘 구별한다.)

1. 파라미터 fine-tune
2. 고차원->저차원 분류해서 데이터 사이의 유사도
3. GAN으로 데이터 늘리기

미니멀리스트 데이터 세트로부터 신뢰성 있는 알고리즘!

데이터셋 훈련에 사용 => 서포트 데이터 / 테스트에 사용 => 쿼리 데이터 = N-way K-shot 문제

=> N은 범주, K는 범주별 데이터 수 = N이 커지면 성능이 낮아지고, K가 커지면 성능이 올라갈 것

=> 일반적인 벤치마크에서는 N은 10개 이하, K는 1개~5개 / 쿼리 데이터는 범주당 15개!

=> 5-way 1-shot -> 정확도 50%, 5-way 5-shot -> 정확도 70%

에피소딕 훈련 방식이 필요 : 샘플링을 다른 범주를 포함시킴 => {고양이, 자동차}, {사과, 귤} ..

테스트 셋은 완전 새로운 시퀀스!

그러면 어떤식으로 학습하냐? -> metric learning(거리 학습), 그래프 신경망 과 관련된 6편!

=> 가중치 업데이트 방식으로는 무조건 overfit! => 대신에 서포트 데이터와 쿼리 데이터 간의 거리(유사도)를 측정! => 쿼리 데이터의 공간상의 위치 중 가장 가까운 서포트 데이터의 클래스로 예측

범주가 같다면 더 가깝게, 다르다면 멀게 학습

이번엔 그래프 신경망 방식!

Active learning: 좋은 데이터셋 => unlabeled dataset predict => 훈련 dataset 증가 => 다시 learn