Relation Classification as Two-way Span-Prediction

Authors: Amir DN Cohen, Shachar Rosenman, Yoav Goldberg

발표자: 구선민

00 Background

Relation Extraction

문장(혹은 문서)에서 entity의 관계를 예측하는 task

- Ex. “Barack Obama was born in Honolulu, Hawaii.”
  → relation classifier는 “born In City”의 관계를 예측

- relation knowledge graphs의 핵심 요소
- 다양한 응용 태스크에 사용(QA, MRC …)

* Deep Residual Learning for Weakly-Supervised Relation Extraction

01 Introduction

Introduction

- relation classification (RC) task는 2개의 엔티티 사이의 binary relation을 중심으로 진행

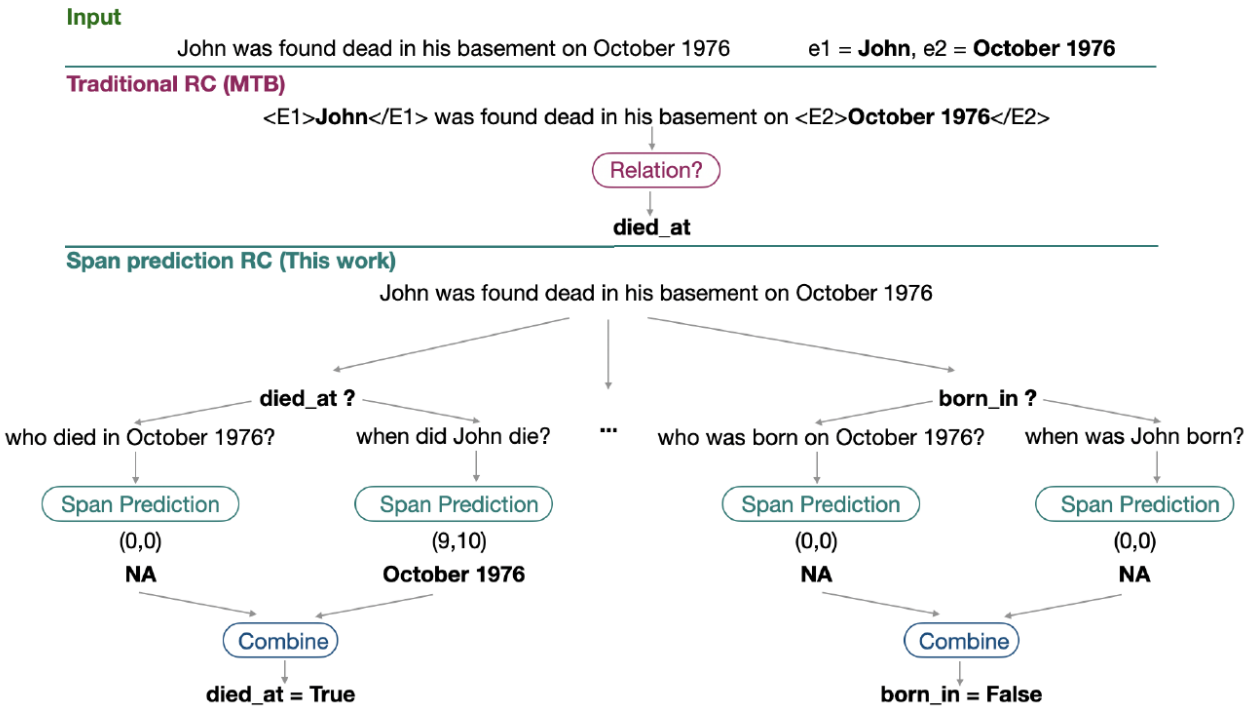
- Corpus를 읽고 text에 따른 entity pair e1, e2를 return
  -> sentence와 2개의 entity가 주어지는(각 엔티티는 span over the sentence) RC task 취급
  -> possible relation R로 분류하거나, 관계 없음" no-relation"

- TACRED 데이터셋
  - 형식 (s, e_1, e_2, r) //s: sentence e1, e2: entity, r: e1과 e2의 semantic relation
  - 한 문장 내에서도 3개 이상 entity 있을 수 있지만 데이터셋에 맞춰 2개의 entity 간의 관계 추출

- 현재의 supervised relation classification(RC) task는 entity pair 사이의 관계를 표현하기 위해 single embedding 사용

- **QA model 처럼 relation을 찾는 span prediction 접근법을 사용함**

Introduction



- RC dataset을 SP 형식으로 바꾸어 사용
- Entity type에 따른 relation type에 대해 2개의 question 생성
- 2개의 질문 중 맞게 answer한 것이 두 Entity의 relation

02 Embedding Classification
vs
Span-Prediction

Embedding Classification vs Span-Prediction

- **Embedding Classification**

$$f_{rc}(c, e_1, e_2) \mapsto r \in R \cup \{\emptyset\}$$

- $(c, e_1, e_{2,r})$
- $c: context$
- $e: spans\ that\ correspond\ to\ head\ and\ tail\ entities$
- $r: relation$

- 같은 relation을 가진 인스턴스는 비슷한 위치에 임베딩 됨

Span Prediction

$$f_{qa}(c, q) \mapsto e_a \in [1..m] \times [1..m]$$

- $(c, q\ e_a)$; $c: context, q: query, e: answer$

$$\arg\max_{e_a} score_{c,q}(e_a)$$

- $score_{c,q}(e_a)$ 를 최댓값으로 만들기 위한 e_a를 구함

03 Method Comparison

Embedding

- 두 methods는 context, two spans, relation/predicate 포함

- RC models은 2개의 span에서 relation으로 분류
- RC의 임베딩은 context와 entities 에 기초

$$h_{qa} = embed(q, c) = embed(r, e_1, c)$$

- SP model은 span과의 관계에서 다른 span으로 분류

- Context와 Question (interest와 entity 1개와의 관계)를 모두 인코딩
- Relation name과 (r, e_1)을 둘러싼 template word가 포함

- (본 연구에서는 BERT로 실험했는데 CLS token과 , SEP token으로 문장 시작과 다른 문장과의 구분 표시)

04 Implications

Implication

Relation type indication for the pretrained model.

- Contextualized embedder에 relation을 포함하여 input을 넣으면 embedder가 특정 relation이 specialize 할 수 있음

- Ex. "Martha gave birth to John last February"
  - 엔티티 John은 2가지 "date of birth", "parents of" relation에 해당할 수 있음

- RC 임베딩의 경우 엔티티에 근거하여 relation을 추정하거나 두 relation에 대한 정보를 유지해야함

- Span Prediction의 경우 임베딩은 relation의 argument로서 1개의 entity에 집중

- 임베딩 단계에서 Specific relation에 집중하여 모든 model에 적용할 수 있음

Implication

Sharing of semantic information.

- Span prediction model은 r, e를 인코딩하는 template 통해 중요한 information을 모델에 전달
  - (1) target relation의 semantic information(ex. relation을 나타내는 question)
  - (2) 다른 relation을 일반화하는데 도움이 될 수 있는 information

- Ex. relation: “born in”, template question: “Who was born in X?”
  relation: “parent of”, template question: “Who is the parent of X?
”

  → Relation은 다르지만 둘 다 엔티티 타입이 “person”, “Who”라는 단어가 공통으로 있기 때문에 모델에 유사성이 전달되어 relation type 간의 일반화하는데 도움

Implication

RC sample	Relation candidates	Question (Reverse Question)	Answer
John was born on **1991**	"Date of birth"	When was John Born?	1991
		(Who was born on 1991?)	John
	"Date of death"	When did john die?	N/A
		(Who died on 1991?)	N/A
Mary is **John**'s employer	"employer of"	Who is employed by Mary?	John
		(Who is John employer?)	Mary
	"siblings"	Who is the sibling of Mary?	N/A
		(Who is the sibling of John?)	N/A
	"parents of"	Who is the child of Mary?	N/A
		(Who is the parent of John?)	N/A

- Reverse Question으로 양방향으로 수행하여 한쪽이 틀리거나 양쪽이 틀리면 no relation
- 단점: entity 주어졌을 때 해당하는 모든 relation을 다 해봐야 하므로 시간이 오래 걸림

05 Reducing RC to span-prediction

Reducing RC to span-prediction

RC instance $(c, e_1, e_2) \mapsto rel$

SP instance $(q = (e_q, rel_q), c) \mapsto e_a$

- $T_{rel}(e)$: 엔티티 'e'를 받아 question을 return
- $ex.\ T_{dob} = "When\ was\ \_\_\_\ born"$ → $T_{dob}(Sam) = "When\ was\ Sam\ born"$

QA1:$(c, When\ was\ Sam\ born?) \mapsto 1991$

QA2:$(c, Who\ was\ born\ in\ 1991?) \mapsto Sam$

- 양방향으로 질문 생성

05 Result

Result

Model	Acc_+	Acc_-	Acc
RC_{BERT}	70.0	64.8	67.1
$SQuAD_{BERT}$	62.9	70.9	67.4
$SP_{token,BERT}$	55.0	75.5	66.4
$SP_{relation,BERT}$	66.6	72.1	69.6
$SP_{question,BERT}$	**72.5**	**75.0**	**73.9**
$SQuAD_{ALBERT}$	71.5	78.8	75.3
$SP_{token,ALBERT}$	80.9	73.2	76.6
$SP_{relation,ALBERT}$	78.2	79.8	79.1
$SP_{question,ALBERT}$	**81.2**	**79.5**	**80.3**

Model	P	R	F_1
$RC_{MTB,BERT}$	-	-	70.1
$SP_{token,BERT}$	63.3	78.4	70.0
$SP_{relation,BERT}$	67.0	76.0	71.2
$SP_{question,BERT}$	**71.1**	**72.6**	**71.8**
$KEPLLER_{RoBERTa+KG}$	72.8	72.2	72.5
$SP_{token,ALBERT}$	72.2	74.6	73.4
$SP_{relation,ALBERT}$	**74.6**	**75.2**	**74.8**
$SP_{question,ALBERT}$	73.3	71.8	72.6

Model	P	R	F_1
$RC_{MTB,BERT}$	-	-	89.2
LiTian (current best)	**94.2**	88.0	91.0
$SP_{token,BERT}$	92.8	88.8	90.7
$SP_{relation,BERT}$	91.9	83.1	87.1
$SP_{question,BERT}$	90.7	**93.2**	**91.9**

- BERT보다 ALBERT의 성능이 더 높음
- RC task를 QA와 유사하게 span-prediction(SP) problem으로 처리하는 것이 더 나은 접근법이라고 주장
- supervised SP objective가 standard classification based objective보다 더 잘 동작한다는 것을 증명