Abstract

KE(Knowledge Embedding)에 대한 open toolkit을 제공.(knowledge graph를 continuous low-dimensional space로 embed)

연산적 효율성과 큰 규모의 knowledge를 representation하는 learning에 중점을 두고있다.

또한 다른 모델과의 합성, 확장에 유용.

OpenKE에 의해 pre-trained된 큰 규모의 KG의 임베딩은 information retrieval, personalized recommendation and question answering등의 작업에 사용.

1. Introduction

현실 세계를 반영하는 structured knowledge를 구성하기 위해 큰 규모의 다양한 KGs가 있다. (WordNet, Freebase and Wikidata).

대부분의 KG들은 triples의 형태로 구성된다. (h,r,t) => h와 t는 head and tail entities를 의미. R은 h와 t사이의 relation을 나타냄. Ex) (Mark Twain PlaceOfBirth Florida)

다양한 knowledge-driven NLP 작업을 강화시키기 위해 풍부한 structured information들이 사용된다. (KG를 계속 효과적으로 구성하면서? = 늘려가는 뜻인가? 이게 뭘까)

KG의 규모와 띄엄띄엄 있는 특성의 제한으로 KG를 distributed representations으로 표현해야한다.  
따라서 KE approach는 entities와 relations를 continuous low-dimensional space로 embed 하도록 제안되어 왔다. (이전까지 연구된 많은 저자들, 모델들,,, 그 중에서도 matrix factorization models, 최근(2020)에는 Probability Calibration)

좋은 성과, 하지만 아직 두 가지의 메인 이슈가 남아있다. 현재 기법은 흩뿌려지고, 덜 체계적이다. = 각각 비일관된다. 다른 한편, 이 모델들은 model validation과 time-consuming에만 초점. = 이러면 실세계에서는 적용하기 힘들어진다. 따라서 성능(효율적이고 효과적인??)이 좋은 embedding model이 필요. = OpenKE

특징으로는 3가지가 있다.

1) data와 memory level에서 같이 통합적으로 사용하는 KE models에 대해 data와 memory를 관리? 복잡한 데이터 처리와 메모리 할당은 필요없다.

2) 알고리즘 level에서 OpenKE는 다양한 구체적인 모델의 수학적 폼을 통합한다. (또한 더 발전을 위한 novel negative sampling strategy 제안..???)

3) computation level에서 OpenKE는 KG를 몇 개의 part로 나눌 수 있고 KE model을 parallel training 가능하게 한다. (GPU에서 모델을 돌리기 위해 TF와 PT도 채택)

또한 잘 알려진 큰 규모의 KG의 embedding은 제공하며(word2vec처럼), 다른 작업에 바로 사용가능. 아키텍쳐 디자인과 implementation에 대해 중점적으로 설명하고, 형식적인 KE 모델의 OpenKE와 함께 했을 때의 평가를 진행할 것이다.

2. background

KG(g)에서 data는 directed graph로써 표현된다.  각각, entity, relation, fact 이다. fact에는 (h,r,t) 단위로 값이 들어가고 (h,r,t) 중 r은 R에, h와 t는 E에 포함된다.

이후 low-dimensional vector로 representation 하게 되면 bold체로 **h,r,t가** 된다.

(h,t)의 entity 쌍과 relation r에 대해 fact (h,r,t)가 embedding 돼 low-dimensional로 표현되는지 결정할 수 있다. 이런 임베딩은 KG에서 knowledge를 이해하고 mining하는 것을 촉진한다.

KE 모델은 각 triple에 S(h,r,t)라는 scoring function을 정의한다. 대부분의 경우 true triples이고, non-existing triples는 false이거나 missing일 수 있다. Dong은 이 문제를 해결 => true triples면 높은 S(h,r,t)를 반환.

Scoring function을 이용해 KE 모델은 margin-based loss를 설정.(entitiy와 relation의 임베딩을 배운다.)



첫 항 = margin > 0 /[]+는 양수 값만 남김 / T’은 non-existing triples(existing triples에서 entity와 relation이 부정확??)

Classification task로 여길 경우 loss function이 달라질 수 있다.



여러 model의 차이는 scoring function이 다른 것이다.(true triples의 구별?) 여러 모델과, openKE와의 통합을 보여줄 것이다.