Motivation

- Fundamental Issues in KB-QA
  1. Given a natural language question, how to identify entity spans of the question?
  2. How to map the question to its corresponding logical predicate?

Solution: Jointly train semantic relations between a question context and logical properties of KB (entities and logical predicates) in the same embedding space.

Relational Components for KB-QA

- Question context (C): represented as n-grams
- Entity type (T): abstract expression of target entities
- Logical predicate (P): canonical form of NL relation phrases

NLE-KB Pair Extraction

- NLE-KB pair: semantically associated tuples for training relational embeddings between NL and KB space
  - <Relation Mention, Predicate> pair (MP)
    - NL-entries: <relation mention, entity span>
      - NLE: "Barack Obama, be born in, Honolulu"
      - KB: "Barack Obama, place_of_birth, Honolulu"
    - KB-triples: <entity type (t), logical predicate (p)>
      - entity type: "person"
      - logical predicate: "place_of_birth"

Joint Relational Embedding Learning

- Construction of training instances
  1. Each NLE-KB pair → multiple training triplets
    - Training triplet \( w = [C, t, p] \): (C: NLE, t and p: KB)
  2. Each training triplet → 3 training pairs
    - Training pairs: \( R = \{C-t, C-p, t-p\} \)

In C, the placeholder "<entity>" for a target entity is left.

Ranking loss-based learning (Winston et al., 2010)

Assumption: similarity scores of observed pairs in the training set should be higher than those of any other pairs

- \( \forall x, y, x' \neq y, Sim(x, y) > 1 + Sim(x, y') \)
- Similarity score: \( Sim(a, b) = Sim(\alpha, \beta) = \mathbb{E}(a)^T \mathbb{E}(b) \)

Embeddings of C, T, and P are trained under the SGD by the above criterion

- \( \forall x, y, x' \neq y, max(0, 1 - Sim(x, y) + Sim(x, y')) \)

KB-QA using Embedding Models

Given a natural language question \( q \) (single-related question),

1. Make all possible decoding triplets \( W^q \), like a training triplet
   - C: n-grams of q (entity span is replaced with "<entity>")
   - t: one of all available entity types via Search API on KB with all string spans in q (candidate entities; s)
   - p: one of all items in \( P \) (candidate logical predicates)
   - \( w_k^q = [C_k^q, t_k^q, p_k^q] \) is directly linked to KB-query \( k \) = \( \{t_k^q, p_k^q, s\} \), any entities on "s" can be potential answers

2. Score \( W^q \) through embedding space
   - Similarities of \( R^q = \{C_k^q - t_k^q, C_k^q - p_k^q, t_k^q - p_k^q\} \) are computed
   - \( Sim_{all}(q, k) = \sum_{p} Sim(q, k) \) normalization
   - \( k(q) = \text{arg max } Sim_{all}(q, k) \) corresponding KB-query

- Multi-related question (# target entities = 2)
  - Heuristic rule: transformed to single-related question
  - If a pre-defined pair of entity types is detected, they are combined into a concatenated entity type
  - The concatenated entity is regarded as one of the candidate entities

Example question: "where is the city of david?"

Experimental Evaluation

- Resource: Satori KB / 4.4 M Wikipedia articles
- Features: 71,310 n-grams (uni-, bi-, tri-) / 990 entity types / 660 logical predicates (72,960 embeddings)
- Embedding learning: dimension=100, learning rate=0.00001
- Evaluation data: publicly released QA data sets
  - Free917: 276 QA-pairs (Cai et al., 2013)
  - WebQuestions: 2,032 QA-pairs (Berant et al., 2013)

Accuracy on evaluation data

- Methods: Free917 / WebQ.
- Accuracy: average of \( F_1 \) scores over all of test questions

Impacts of relationship types

- Accuracy of the other methods are from their papers

Problems to be solved

- Complex questions requiring multiple stages to detect their target entities
- Uncommon questions consisting of rare n-grams