Question Answering with Subgraph Embeddings

A. Bordes, S. Chopra, J. Weston
(presented by Karel Ha)

facebook®

Open-domain question answering

Definition (Open QA)

the task to automatically answer questions asked in natural language **on any topic** or **in any domain**

Nowadays: lookup in large scale structured knowledge bases (KBs)

Open-domain question answering

Definition (Open QA)

the task to automatically answer questions asked in natural language **on any topic** or **in any domain**

Nowadays: lookup in large scale structured knowledge bases (KBs)

Two main approaches, based on:

Open-domain question answering

Definition (Open QA)

the task to automatically answer questions asked in natural language **on any topic** or **in any domain**

Nowadays: lookup in large scale structured knowledge bases (KBs)

Two main approaches, based on:

- information retrieval (Yao and Van Durme, '14)
- semantic parsing (Berant et al., '13; Berant and Liang, '14)

Open-domain question answering

Definition (Open QA)

the task to automatically answer questions asked in natural language **on any topic** or **in any domain**

Nowadays: lookup in large scale structured knowledge bases (KBs)

Two main approaches, based on:

- information retrieval (Yao and Van Durme, '14)
- semantic parsing (Berant et al., '13; Berant and Liang, '14)

A lot of human supervision: lexicons, grammars, KB schema ...

Embedding model

. . . learns representations as **low-dimensional vectors** of words and KBs elements

Embedding model

. . . learns representations as **low-dimensional vectors** of words and KBs elements

In particular, **subgraph embeddings**:

- 👍 richer representation of the answers (QA path, surrounding subgraph of the KB)

Embedding model

...learns representations as **low-dimensional vectors** of words and KBs elements

In particular, **subgraph embeddings**:

- richer representation of the answers (QA path, surrounding subgraph of the KB)
- less human supervision

Embedding model

...learns representations as **low-dimensional vectors** of words and KBs elements

In particular, **subgraph embeddings**:

- richer representation of the answers (QA path, surrounding subgraph of the KB)
- less human supervision
- ability to answer more complicated questions (indirect answers)

Embedding model

. . . learns representations as **low-dimensional vectors** of words and KBs elements

In particular, **subgraph embeddings**:

- richer representation of the answers (QA path, surrounding subgraph of the KB)
- less human supervision
- ability to answer more complicated questions (indirect answers)
- more sophisticated inferences

Freebase

the knowledge base of general facts

Freebase

the knowledge base of general facts

Database of 14 million triplets:

FREEBASE

the knowledge base of general facts

Database of 14 million triplets:

- `subject`
- `object`
- connected by `type1.type2.predicate`

Freebase

the knowledge base of general facts

Database of 14 million triplets:

- `subject`
- `object`
- connected by `type1.type2.predicate`

Turned automatically into questions-answer pairs for training:
“What is the `predicate` of the `type2 subject`? (`object`)”

Freebase

the knowledge base of general facts

Database of 14 million triplets:

- subject
- object
- connected by type1.type2.predicate

Turned automatically into questions-answer pairs for training:
"What is the predicate of the type2 subject? (object)"
"What is the nationality of the person barack_obama? (united_states)"

WebQuestions

the dataset of 5810 QA pairs built from Freebase

WebQuestions

the dataset of 5810 QA pairs built from Freebase

Questions from:

Google Suggest

Answers from:

amazon mechanical turk

beta

WebQuestions

the dataset of 5810 QA pairs built from Freebase
Questions from:

Google Suggest

Answers from:

amazon mechanical turk

One Freebase entity identified in each (natural) question using string matching:
"What degrees did Barack Obama get? (`bachelor_of_arts`, `juris_doctor`)"

ClueWeb Extractions

- provided by (Lin et al. '12)
- 2 million extractions

ClueWeb Extractions

- provided by (Lin et al. '12)
- 2 million extractions
- (subject, "text string", object)
- "Where barack_obama was allegedly bear in? (hawaii)"

Paraphrases

There are issues with automagically generated questions:

- semi-automatic wording
- rigid syntax
- often unnatural
- ...

Paraphrases

There are issues with automagically generated questions:

- semi-automatic wording
- rigid syntax
- often unnatural
- ...

⇒ pairs of question **paraphrases**:

- from WikiAnswers (users have option to tag rephrasings of questions)
- 2 million distinct questions
- 350,000 paraphrase clusters

Scoring function

Let us have a question q and a candidate answer a.
Goal: to learn the scoring function

$$S(q, a) = f(q)^\top g(a)$$

So that: high score if a is the correct answer; low score otherwise

Scoring function

Let us have a question q and a candidate answer a.
Goal: to learn the scoring function

$$S(q, a) = f(q)^\top g(a)$$

So that: high score if a is the correct answer; low score otherwise
Where:

- k is the dimension of the embedding space
- N_W is the number of words, N_S the number of entities and relations, $N = N_W + N_S$

Scoring function

Let us have a question q and a candidate answer a.

Goal: to learn the scoring function

$$S(q, a) = f(q)^\top g(a)$$

So that: high score if a is the correct answer; low score otherwise

Where:

- k is the dimension of the embedding space
- N_W is the number of words, N_S the number of entities and relations, $N = N_W + N_S$
- $\mathbf{W} \in \mathbb{R}^{k \times N}$, where the column $\mathbf{W}_{*,i}$ is the embedding of the i-th word, entity or relation

Scoring function

Let us have a question q and a candidate answer a.
Goal: to learn the scoring function

$$S(q, a) = f(q)^\top g(a)$$

So that: high score if a is the correct answer; low score otherwise
Where:

- k is the dimension of the embedding space
- N_W is the number of words, N_S the number of entities and relations, $N = N_W + N_S$
- $\mathbf{W} \in \mathbb{R}^{k \times N}$, where the column $\mathbf{W}_{*,i}$ is the embedding of the i-th word, entity or relation
- vectors $\varphi(q) \in \mathbb{N}_0^N$ indicates how many times each word occurs in the question
- $\psi(a) \in \mathbb{N}_0^N$ is the vector representation of the answer (next slide)

Scoring function

Let us have a question q and a candidate answer a.
Goal: to learn the scoring function

$$S(q, a) = f(q)^\top g(a)$$

So that: high score if a is the correct answer; low score otherwise
Where:

- k is the dimension of the embedding space
- N_W is the number of words, N_S the number of entities and relations, $N = N_W + N_S$
- $\mathbf{W} \in \mathbb{R}^{k \times N}$, where the column $\mathbf{W}_{*,i}$ is the embedding of the i-th word, entity or relation
- vectors $\varphi(q) \in \mathbb{N}_0^N$ indicates how many times each word occurs in the question
- $\psi(a) \in \mathbb{N}_0^N$ is the vector representation of the answer (next slide)
- $f(q) = \mathbf{W}\varphi(q)$ and $g(a) = \mathbf{W}\psi(a)$ are embeddings into $\mathbb{R}^k$

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between
  - `(barack_obama, people.person.place_of_birth, honolulu)`

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between
  - `(barack_obama, people.person.place_of_birth, honolulu)`
  - `(barack_obama, people.person.place_of_birth, location.location.containedby, hawaii)`

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between
  - `(barack_obama, people.person.place_of_birth, honolulu)`
  - `(barack_obama, people.person.place_of_birth, location.location.containedby, hawaii)`
- **Subgraph Representation**
  - path representation + the entire subgraph of entities connected to the candidate answer entity

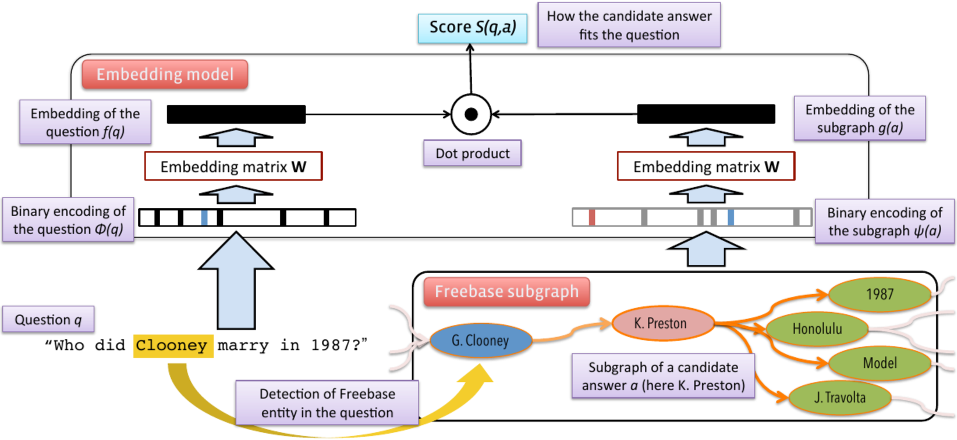
Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between
  - `(barack_obama, people.person.place_of_birth, honolulu)`
  - `(barack_obama, people.person.place_of_birth, location.location.containedby, hawaii)`
- **Subgraph Representation**
  - path representation $+$ the entire subgraph of entities connected to the candidate answer entity
  - double the size for entities and relations: $N = N_W + 2N_S$

Representing Candidate Answers

- **Single Entity**
  - 1-of-N_S coded vector with 1 corresponding to the entity of the answer, and 0 elsewhere
- **Path Representation**
  - 3-of-N_S or 4-of-N_S coded vector: 1-hop or 2-hop path from the question entity to the answer entity using the relation types (but not entities) in-between
  - `(barack_obama, people.person.place_of_birth, honolulu)`
  - `(barack_obama, people.person.place_of_birth, location.location.containedby, hawaii)`
- **Subgraph Representation**
  - path representation + the entire subgraph of entities connected to the candidate answer entity
  - double the size for entities and relations: $N = N_W + 2N_S$
  - C entities connected to the answer entity via D relations: either $(3 + C + D)$-of-N_S or $(4 + C + D)$-of-N_S coded vector

Overview


Score S(q,a)
How the candidate answer fits the question
Embedding model
Embedding of the question f(q)
Dot product
Embedding of the subgraph g(a)
Embedding matrix W
Embedding matrix W
Binary encoding of the question Φ(q)
Binary encoding of the subgraph ψ(a)
Freebase subgraph
Question q
"Who did Clooney marry in 1987?"
G. Clooney
K. Preston
1987
Honolulu
Model
J. Travolta
Subgraph of a candidate answer a (here K. Preston)
Detection of Freebase entity in the question

Training and Ranking Loss Function

Let $\mathcal{D} = \{(q_i, a_i) : i = 1, \ldots, |\mathcal{D}|\}$ be the training set of QA pairs.

Training and Ranking Loss Function

Let $\mathcal{D} = \{(q_i, a_i) : i = 1, \ldots, |\mathcal{D}|\}$ be the training set of QA pairs.
Goal: to minimize

$$\sum_{i=1}^{|\mathcal{D}|} \sum_{\overline{a} \in \overline{\mathcal{A}}(a_i)} \max\{0, (S(q_i, \overline{a}) + m) - S(q_i, a_i)\}$$

So that: the score of a question paired with a correct answer is greater than with any incorrect answer $\overline{a}$ by at least m.

Training and Ranking Loss Function

Let $\mathcal{D} = \{(q_i, a_i) : i = 1, \ldots, |\mathcal{D}|\}$ be the training set of QA pairs.
Goal: to minimize

$$\sum_{i=1}^{|\mathcal{D}|} \sum_{\overline{a} \in \overline{\mathcal{A}}(a_i)} \max\{0, (S(q_i, \overline{a}) + m) - S(q_i, a_i)\}$$

So that: the score of a question paired with a correct answer is greater than with any incorrect answer $\overline{a}$ by at least m.
Where:

- margin parameter m, the set of incorrect candidates $\overline{\mathcal{A}}$
- $\overline{a}$ sampled from $\overline{\mathcal{A}}$

Training and Ranking Loss Function

Let $\mathcal{D} = \{(q_i, a_i) : i = 1, \ldots, |\mathcal{D}|\}$ be the training set of QA pairs.
Goal: to minimize

$$\sum_{i=1}^{|\mathcal{D}|} \sum_{\overline{a} \in \overline{\mathcal{A}}(a_i)} \max\{0, (S(q_i, \overline{a}) + m) - S(q_i, a_i)\}$$

So that: the score of a question paired with a correct answer is greater than with any incorrect answer $\overline{a}$ by at least m.

Where:

- margin parameter m, the set of incorrect candidates $\overline{\mathcal{A}}$
- $\overline{a}$ sampled from $\overline{\mathcal{A}}$
  - 50% of time: another entity connected to the question entity (i.e., through other candidate paths)
  - 50% of time: random answer entity

Training and Ranking Loss Function

Let $\mathcal{D} = \{(q_i, a_i) : i = 1, \ldots, |\mathcal{D}|\}$ be the training set of QA pairs.
Goal: to minimize

$$\sum_{i=1}^{|\mathcal{D}|} \sum_{\overline{a} \in \overline{\mathcal{A}}(a_i)} \max\{0, (S(q_i, \overline{a}) + m) - S(q_i, a_i)\}$$

So that: the score of a question paired with a correct answer is greater than with any incorrect answer $\overline{a}$ by at least m.
Where:

- margin parameter m, the set of incorrect candidates $\overline{\mathcal{A}}$
- $\overline{a}$ sampled from $\overline{\mathcal{A}}$
  - 50% of time: another entity connected to the question entity (i.e., through other candidate paths)
  - 50% of time: random answer entity
- optimization:
  - stochastic gradient descent
  - multi-threaded
  - columns of **W** are inside the unit ball: $\forall i : ||\mathbf{W}_{*,i}|| \leq 1$

Multitask Training of Embeddings

Synthethic questions inadequately cover the syntax of natural language!

Multitask Training of Embeddings

Synthethic questions inadequately cover the syntax of natural language!

$\Rightarrow$ Multi-task the training with the task of **paraphrase prediction**!

Multitask Training of Embeddings

Synthethic questions inadequately cover the syntax of natural language!

$\Rightarrow$ Multi-task the training with the task of **paraphrase prediction**!

- scoring function $S_{prp}(q_1, q_2) = f(q_1)^\top f(q_2)$

Multitask Training of Embeddings

Synthethic questions inadequately cover the syntax of natural language!

$\Rightarrow$ Multi-task the training with the task of **paraphrase prediction**!

- scoring function $S_{prp}(q_1, q_2) = f(q_1)^\top f(q_2)$
- same embedding matrix $\mathbf{W}$

Multitask Training of Embeddings

Synthethic questions inadequately cover the syntax of natural language!

$\Rightarrow$ Multi-task the training with the task of **paraphrase prediction**!

- scoring function $S_{prp}(q_1, q_2) = f(q_1)^\top f(q_2)$
- same embedding matrix $\mathbf{W}$
- negative samples taken from different paraphrase clusters

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

- the whole KB (slow and imprecise!)

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

- the whole KB (slow and imprecise!)
- C_1: only Freebase triplets involving the question entity (simple factual questions, 1-hop paths)

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

- the whole KB (slow and imprecise!)
- C_1: only FREEBASE triplets involving the question entity (simple factual questions, 1-hop paths)
- C_2: additional triplets at 2-hop distance from the question entity (i.e., neighbors of neighbors)

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

- the whole KB (slow and imprecise!)
- C_1: only FREEBASE triplets involving the question entity (simple factual questions, 1-hop paths)
- C_2: additional triplets at 2-hop distance from the question entity (i.e., neighbors of neighbors)
  - not all the quadruplets (too large set), but predictions made in turns, using best-first search heuristics
  - rank KB's relations using $S(q, a)$

Inference

The trained model predicts the answer by

$$\widehat{a} = \arg\max_{a' \in \mathcal{A}(q)} S(q, a')$$

where $\mathcal{A}_q$ is the candidate answer set for each question, chosen as:

- the whole KB (slow and imprecise!)
- C_1: only FREEBASE triplets involving the question entity (simple factual questions, 1-hop paths)
- C_2: additional triplets at 2-hop distance from the question entity (i.e., neighbors of neighbors)
  - not all the quadruplets (too large set), but predictions made in turns, using best-first search heuristics
  - rank KB's relations using $S(q, a)$
  - top 10 relations $\Rightarrow$ only 2-hop paths containing them
  - 1-hop triplets weighted by value 1.5

Multiple answers

"Who are David Beckham's children?" $\Rightarrow$ a whole set of answers, all on the path:

(david_beckham, people.person.children, *)

Multiple answers

"Who are David Beckham's children?" $\Rightarrow$ a whole set of answers, all on the path:

`(david_beckham, people.person.children, *)`

Vector representing multiple answers is the average of sub-answers:

$$\psi_{all}(a') = \frac{1}{|a'|} \sum_{a'_j \in a'} \psi(a'_j)$$

Results on the WebQuestions test set

Method	P@1 (%)	F1 (Berant)	F1 (Yao)
Baselines			
(Berant et al., 2013)	–	31.4	–
(Bordes et al., 2014b)	31.3	29.7	31.8
(Yao and Van Durme, 2014)	–	33.0	42.0
(Berant and Liang, 2014)	–	39.9	43.0
Our approach			
Subgraph & $\mathcal{A}(q) = C_2$	**40.4**	39.2	43.2
Ensemble with (Berant & Liang, 14)	–	**41.8**	**45.7**
Variants			
Without multiple predictions	**40.4**	31.3	34.2
Subgraph & $\mathcal{A}(q) =$ All 2-hops	38.0	37.1	41.4
Subgraph & $\mathcal{A}(q) = C_1$	34.0	32.6	35.1
Path & $\mathcal{A}(q) = C_2$	36.2	35.3	38.5
Single Entity & $\mathcal{A}(q) = C_1$	25.8	16.0	17.8

Conclusion

Pros of subgraph embeddings:

- training uses only QA pairs and the knowledge base

Conclusion

Pros of subgraph embeddings:

- training uses only QA pairs and the knowledge base
- low-dimension embeddings
- exploiting richer structure of the answers, which is provided by their local neighborhood in the KB graph

Conclusion

Pros of subgraph embeddings:

- training uses only QA pairs and the knowledge base
- low-dimension embeddings
- exploiting richer structure of the answers, which is provided by their local neighborhood in the KB graph
- training for question paraphrasing task at the same time
- promising performance on the WebQuestions benchmark

Thank you!

Thank you!

Time for human Question Answering!