Conversational Word Embedding for Retrieval-based Dialog System

Wentao Ma, Yiming Cui, Ting Liu, Dong Wang, Shijin Wang, Guoping Hu

ACL 2020

State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Research Center for SCIR, Harbin Institute of Technology
iFLYTEK AI Research (Hebei)
What is that animal?

It is an elephant

Why is its nose so long?

Because it needs the nose to drink water

A Conversation between a father and his son in a zoo
Motivation

Language habits

Conversation

Common sense

What is that animal?

It is an elephant

Why is its nose so long?

Because it needs the nose to drink water

Why

Because

Why

Because

Animal

Elephant

Long nose

Drink water

Because
Motivation

• Human conversations contain many types of information, e.g., common sense, language habits and knowledge.

• **cross-sentence**: exist in conversation pair instead of single sentence

• **asymmetric**: some language habits are directional, such as
  • ‘why’ → ‘because’,
  • ‘congratulation’ → ‘thanks’
Related works

- **Word representation methods**
  - **Static word embedding**: Word2vec, GloVe, fastText…
  - **Contextual word embedding**: ELMo, BERT, XLNet…

![Embedding Matrix](image1)

![Pretraining Model](image2)
Related works

- Retrieval-based Dialog System
- Single-turn Response Selection
- Multi-turn Response Selection

Single-turn $g(Q,R)$

Multi-turn $g(C,Q,R)$
Motivation

• Previous word embedding methods for conversation
  • **Single sentence**: the semantic correlation beyond a single sentence is missing
  • **Single vector space**: map the post and reply into the same vector space, which leads the reply with repeated words is easy to be selected

Do you know the animal?
I don’t know
Why you can’t know this?
I don’t know either
Contribution

PR-Embedding: learn conversational word embedding from conversation pairs in two different vector spaces.
Notation

Vocabulary

\[ V^p := \{ v_1^p, v_2^p, \ldots, v_s^p \} \]

Embedding Matrix

\[ E_p = \begin{array}{cccc}
  v_1^p \\
v_2^p \\
v_3^p \\
  \vdots \\
v_s^p \\
\end{array} \]

Sequence

\[ P = (p_1, \ldots, p_m) \]

Post

Reply

\[ V^r := \{ v_1^r, v_2^r, \ldots, v_s^r \} \]

\[ E_r = \begin{array}{cccc}
  v_1^r \\
v_2^r \\
v_3^r \\
  \vdots \\
v_s^r \\
\end{array} \]

\[ R = (r_1, \ldots, r_n) \]
Model

Post:  P_hi  P_,  P_where  P_are  P_you  P_from

Reply:  R_i  R_am  R_from  R_alabama  R_,  R_how  R_about  R_you

Word-level Learning

Sentence-level Learning

Model

P-Embedding  R-Embedding
Model

Word-level Learning

Post: P_hi P_, P_where P_are P_you P_from

Reply: R_i R_am R_from R_alabama R_ R_how R_about R_you
How to generate the *cross-sentence* co-occurrence window?
Model

Word-level Learning

![Word-level Co-occurrence Diagram]
## Model

### Word-level Learning

**Word-level Co-occurrence**

\[
\begin{array}{cccccccc}
V^p_1 & V^p_2 & \cdots & V^p_s & V^r_1 & V^r_2 & \cdots & V^r_s \\
V^p_1 & & & & & & & \\
V^p_2 & & & & & & & \\
\vdots & & & & & & & \\
V^s_p & & & & & & & \\
V^r_1 & & & & & & & \\
V^r_2 & & & & & & & \\
\vdots & & & & & & & \\
V^r_s & & & & & & & \\
\end{array}
\]

**Embedding Matrix** $E_p'$, $E_r'$

\[
\begin{align*}
w_i^T \tilde{w}_k + b_i + \tilde{b}_k &= log(X_{ik})
\end{align*}
\]
Sentence-level Learning

Model
Model

Sentence-level Learning

\[ g(Q, R) \]

Gradient

CNN Max-Pooling

Dot

Gradient

Training

\[ E_p' \]

\[ E_r' \]

\[ P_{hi} \quad P_{,} \quad P_{where} \quad P_{are} \quad P_{you} \quad P_{from} \quad R_{i} \quad R_{am} \quad R_{from} \quad R_{alabama} \quad R_{,} \quad R_{how} \quad R_{about} \quad R_{you} \]
Model

Sentence-level Learning

Gradient

$g(Q, R)$

CNN

Max-Pooling

Gradient

Dot

$E_p'$

PR-Embedding

$E_r'$

$P_{hi}$ $P_$ $P_{where}$ $P_{are}$ $P_{you}$ $P_{from}$ $R_i$ $R_am$ $R_{from}$ $R_{alabama}$ $R_$ $R_{how}$ $R_{about}$ $R_{you}$
Experiment

• Datasets
  • **PersonaChat dataset (Zhang et al., 2018)**
    • English, multi-turn conversation dataset with profile
    • Train/Dev/Test: 133.5k/15.7k/15.1k utterance
    • Evaluation Metrics: hit@k
  • **In-house conversation dataset**
    • Chinese, single-turn conversation dataset
    • Test: 935 posts and 12,767 candidate replies (label with ‘good, middle, bad’)
      Train: 1.07 million pairs after cleaning, from Baidu Zhidao
    • Evaluation Metrics: NDCG, P@1
Experiment

• Result on PersonaChat
  • Single-turn task: compare the embeddings based on BOW (bag-of-words, the average of all word embeddings), only use the current query for prediction
  • Multi-turn task: compare the embeddings based on a neural network KVMemnn, use all the context for prediction

|                      | hits@1 | hits@5 | hits@10 |
|----------------------|--------|--------|---------|
| GloVe<sub>train</sub> | 12.6   | 39.6   | 63.7    |
| GloVe<sub>emb</sub>   | 18.0   | 44.6   | 66.9    |
| BERT<sub>emb</sub>    | 15.4   | 41.0   | 62.9    |
| Fasttext<sub>emb</sub> | 17.8   | 44.9   | 67.2    |
| PR-Embedding          | 22.4   | 60.0   | 81.1    |
| IR baseline†          | 21.4   | -      | -       |
| Starpace†             | 31.8   | -      | -       |
| Profile Memory†       | 31.8   | -      | -       |
| KVMemnn               | 32.3   | 62.0   | 79.2    |
| +PR-Embedding         | 35.9   | 66.1   | 82.6    |
| KVMemnn (GloVe)       | 36.8   | 68.1   | 83.6    |
| +PR-Embedding         | **39.9** | **72.4** | **87.0** |
Experiment

• Result on In-house dataset
  • Single-turn task, compare with GloVe and the public embedding of DSG.
  • P@1(s): only use the candidate reply labeled with ‘good’ as true
• Ablation study
  • w/o PR: change the two vector spaces with the single one, just as the previous method
  • w/o SLL: remove the sentence-level learning

|                  | NDCG | NDCG@5 | P@1  | P@1(s) |
|------------------|------|--------|------|--------|
| GloVe$_{train}$  | 69.97| 48.87  | 51.23| 33.48  |
| DSG$_{emb}$      | 70.82| 50.45  | 52.19| 35.61  |
| BERT$_{emb}$     | 70.06| 48.45  | 51.66| 35.08  |
| PR-Emb           | 74.79| 58.16  | 62.03| 45.99  |
| w/o PR           | 70.68| 50.60  | 50.48| 35.19  |
| w/o SLL          | 71.65| 52.03  | 53.48| 40.86  |
Analysis

• Nearest tokens
  • Four nearest tokens for the three selected words in the whole vector space
  • For PR-Embedding, we **select the words from the post vocabulary** and give the nearest words both in post and reply space

| WHY | GloVe | P-Emb | R-Emb | Global | P-Emb | R-Emb | Global | P-Emb | R-Emb |
|-----|-------|-------|-------|--------|-------|-------|--------|-------|-------|
| why | why   | because| thanks | thanks | welcome | congrats | congrats | thank |
| know| understand | matter | thank  | asking | problem |                | ah | thanks |
| guess| oh     | idea   | fine   | thank | today   | goodness   |    |   |
| so  | probably | reason | asking | good | bill    |    |   | appreciate |


Summary

• We proposed a conversational word embedding method PR-Embedding, which is learned from conversational pairs in two different spaces;

• We introduce the word alignment model from SMT to generate the cross-sentence window, and train the embedding in word and sentence level;

• The experimental results shows PR-Embedding can help the models select better reply by catching the information among the pairs.
Thank you!

wtma@iflytek.com

https://github.com/wtma/PR-Embedding