NLPCC

A Short Texts Matching Method Using Shallow Features and Deep Features

Longbiao Kang, Baotian Hu, Xiangping Wu, Qingcai Chen, Yan He

Harbin Institute of Technology Shenzhen Graduate School

Outline

1. Short Text Conversation
2. Retrieval-based Short Text Conversation
3. Matching Short Texts Using Multi-level Features
4. Experiment
5. Summary

1/ Short Text Conversation

Short Text Conversation： a short text-matching task. For a given short text such as a post, this task aims to find a massive suitable response from the candidate set.

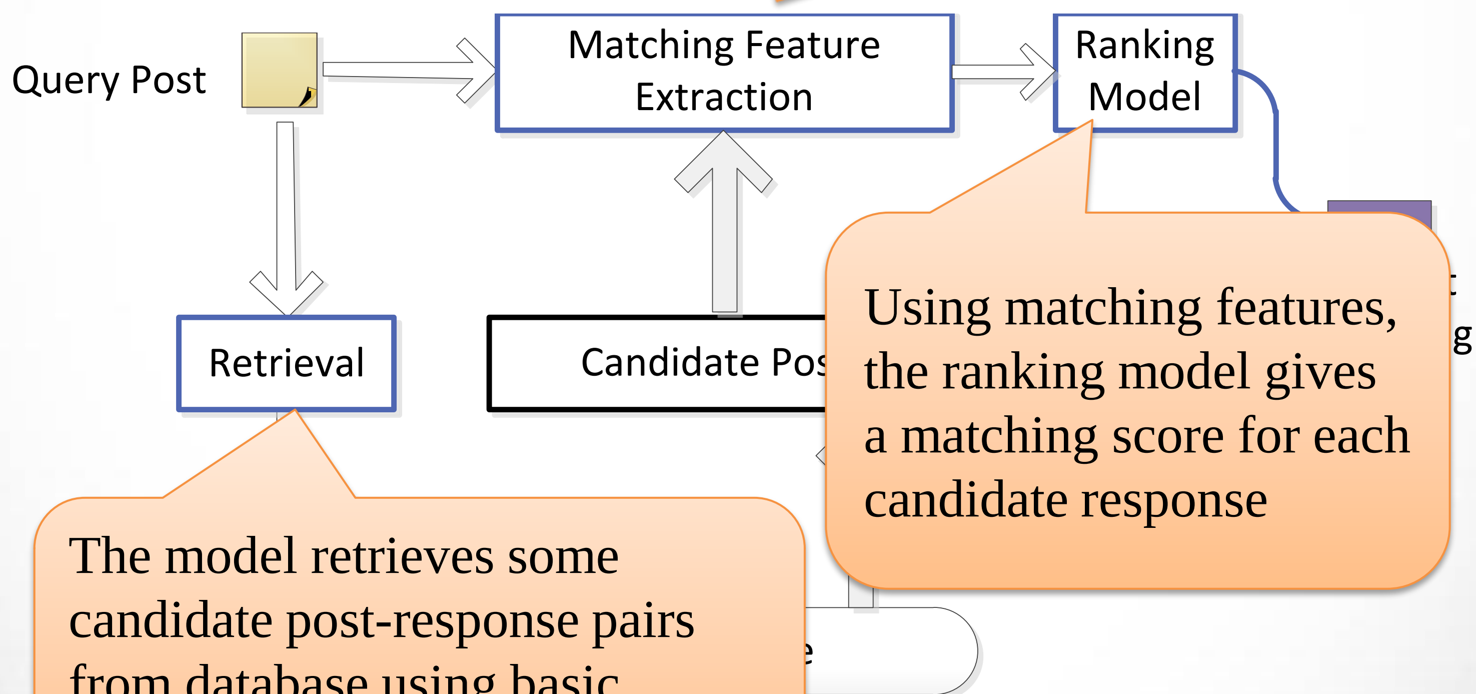
P: 深圳 今天 天气 怎么样？ How is the weather like today in Shenzhen?
R+: 深圳 现在 正 大雨磅礴 。 It's pouring down in torrents now in Shenzhen.
R1-: 在 深圳 过的 怎么样 ？ How is everything going in Shenzhen?
R2-:今天 上海 天气 不错哦 。 The weather in Shanghai is very good today.

- Focusing on the research of short text semantic matching
- It’s a simplified task of modeling a complete dialogue session such as Turing test

2

Retrieval-base

Given a query post q, generate matching features $\{\Phi_i(q,(p,r)), i \in \Omega\}$for each candidate post-response pair (p, r)



Using matching features, the ranking model gives a matching score for each candidate response

The model retrieves some candidate post-response pairs from database using basic retrieval models

3/

Matching Short Texts Using Multi-level Features

- **Multi-level Matching Features**
  - Shallow Matching Features
    - Similarity
    - Latent Semantic Similarity
    - BM25 Score (new work)
  - Deep Matching Features
    - Based on word embedding
    - Based on sentence2vec (new work)
  - Rule-based Matching Features

3 Matching Short Texts Using Multi-level Features

Shallow Features

Deep Features

Rule-based Features

Ranking model

given a post q and a candidate post-response pair (p, r)

- Similarity: vector space model with TF-IDF weights
  - ✓ Similarity between q and r: $\Phi_1(q,(p,r)) = S_{cos}(q,r)$
  - ✓ Similarity between q and p: $\Phi_2(q,(p,r)) = S_{cos}(q,p)$
- Latent Semantic Similarity: Latent semantic indexing (LSI) model
  - ✓ Latent Semantic Similarity between q and r:
    $$\Phi_3(q,(p,r)) = S_{cos}(q_{lsi}, r_{lsi})$$
  - ✓ Latent Semantic Similarity between q and p:
    $$\Phi_4(q,(p,r)) = S_{cos}(q_{lsi}, p_{lsi})$$
- New work: BM25 score of (q, r) and (q, p)

3

Matching Short Texts Using Multi-level Features

Shallow Features

Deep Features

Rule-based Features

Ranking model

A matching model based on neural networks and word embedding

- given a post q and a candidate post-response pair (p, r)

  1. choose the top m words in q by TF-IDF weights

  $$q' = (w_1\ w_2\ w_3\ \dots\ w_m)$$

  2. choose the top n words in r by TF-IDF weights

  $$r' = (w_1\ w_2\ w_3\ \dots\ w_n)$$

  3. convert them into a set of vectors

  $$Q' = (v_1\ v_2\ v_3\ \dots\ v_m)$$
  $$R' = (v_1\ v_2\ v_3\ \dots\ v_n)$$

  4. get a correlation matrix with size $m*n$

  $$M_{cor}' = \{S_{cos}(v_i, v_j) \mid v_i \in Q, v_j \in R\}$$

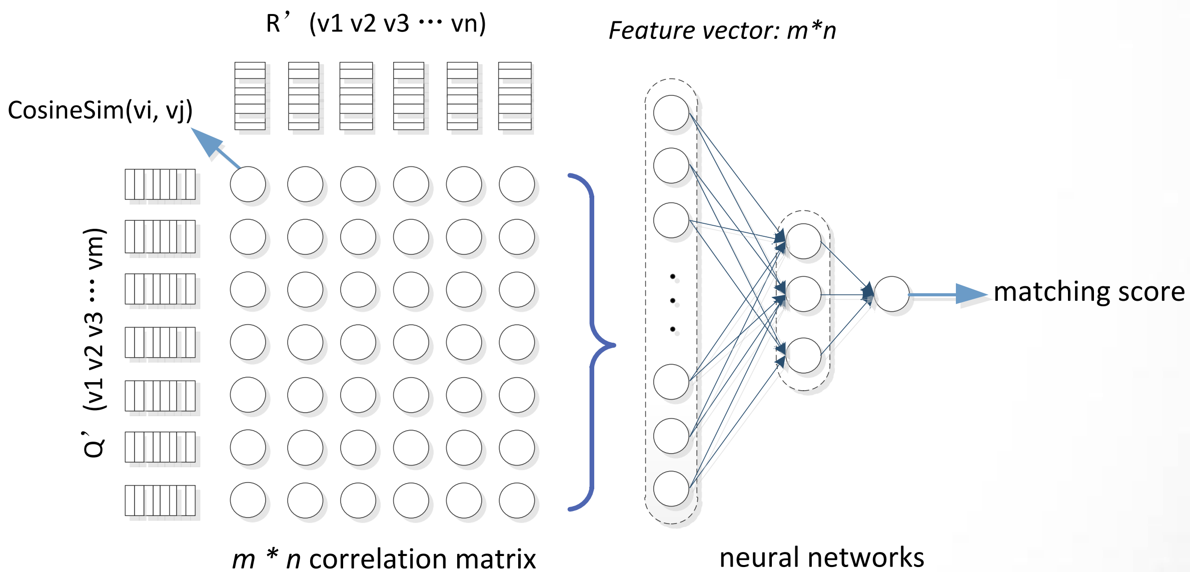
3/ Matching Short Texts Using Multi-level Features

Shallow Features

Deep Features

Rule-based Features

Ranking model



$$\Phi_7(q,(p,r)) = S_{w2v}(q,r)$$

3/ Matching Short Texts Using Multi-level Features

Shallow Features

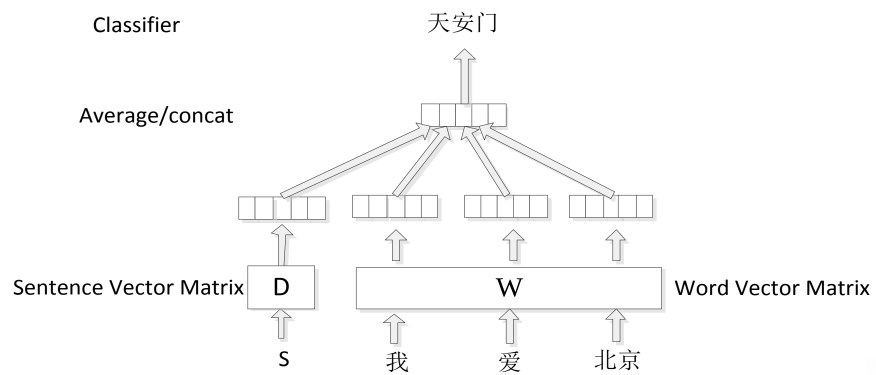
Deep Features

Rule-based Features

Ranking model

New work: matching feature based on distributed representations of sentences (sentence2vec)

- The training model of sentence2vec is similar to word embedding



- given a post q and a candidate post-response pair (p, r)
  - ✓ sentence similarity between q and r:
    $$\Phi_8(q,(p,r)) = S_{s2v}(q,r) = S_{cos}(q_v,r_v)$$
  - ✓ sentence similarity between q and p:
    $$\Phi_9(q,(p,r)) = S_{s2v}(q,p) = S_{cos}(q_v,p_v)$$

3 Matching Short Texts Using Multi-level Features

Shallow Features

Deep Features

Rule-based Features

Ranking model

Matching Features	Definition
$\Phi_{10}(q,(p,r))$	This feature measures whether post q and response r have same entity words.
$\Phi_{11}(q,(p,r))$	This feature measures whether post q and response r have same number such as date and money.
$\Phi_{12}(q,(p,r))$	This feature indicates the length of the longest common string between a post and a response.
$\Phi_{13}(q,(p,r))$	This feature indicates the ratio of the length of post q to the length of response r.

3/ Matching Short Texts Using Multi-level Features

Shallow Features

Deep Features

Rule-based Features

Ranking model

- The ranking model learned a linear score function by RankSVM.

$$Score(q,(p,r)) = \sum_{i \in \Omega} \omega_i \Phi_i(q,(p,r))$$

- Here $\Phi_i(q,(p,r))$ stands for the acquired matching feature and w_i is the weight of $\Phi_i(q,(p,r))$ to be learned.

4/ Experiment

Dataset

Benchmark

Performance

- The dataset of short-text conversation based on the real-world instances from Sina Weibo, which is published by Wang, et al.

Dataset	Use	Size
Unlabeled post-response pairs	For training matching models	Posts: 38,016 Responses: 618,104 Post-response pairs: 618,104
Labeled post-response pairs	For testing	Posts: 422 Responses: 12,402 Responses for each test post: 30
Origin Weibo dataset	For training word embedding	Posts: 33,405,212 Size: 6 GB Vocabulary Size: 10,214 Dimension: 100

4 Experiment

Dataset

Benchmark

Performance

- **MAP:** Mean Average Precision

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$$

- **P@1:** the precision of the top response returned by model

$$P@1 = \frac{|matched(top\ 1)|}{|Q|}$$

4 Experiment

Dataset

Benchmark

Performance

➢ There are four competitor methods:

1. Baseline: A retrieval-based model (Wang, et al. EMNLP 2013)
2. DeepMatch: A deep architecture for matching short texts (Lu and Li. NLPS 2013)

Two newly added models by Ji, et al.

3. TransLM: Translation-based Language Model
4. TopicWord: Topic-Word Model

4 Experiment

Dataset

Benchmark

Performance

- Our early work in paper:

Model and Features	P@1
Retrieval-based Response Model	0.574
DeepMatch	0.424
Shallow Features	0.577
Shallow Features + Deep Features	0.612
All Features	**0.637**

- When combined with deep features, the performance significantly outperforms baseline model.
- Deep features can capture the deep semantic relevance information between two text objects.
- Rule-based features are necessary for some special cases.

4 Experiment

Dataset

Benchmark

Performance

➢ New work after adding BM25, Sentence2vec and new competitor methods.

Model and Features	MAP	P@1
Baseline	0.621	0.574
Baseline + DeepMatch	0.628	0.587
Baseline + DeepMatch + TransLM	0.643	0.625
Baseline + DeepMatch + TransLM + TopicWord	0.654	0.637
Shallow Features	0.593	0.602
Deep Features	0.521	0.531
Shallow Features + Deep Features	0.636	0.640
Shallow Features + Rule-based Features	0.619	0.612
Deep Features + Rule-based Features	0.609	0.610
Shallow + Deep + Rule-based Features	**0.658**	**0.651**

5 Summary

- Short Text Conversation is a challenging problem and current main approach is retrieval-based method.
- Deep features cover rich semantic relevance information between post and response, which the shallow features cannot capture.
- More studies are necessary on matching models based on word embedding and sentence2vec.
- More labeled post-response pairs are required for building more powerful matching models.

References

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. A dataset for research on short-text conversations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP ’13, pages 935–945. ACL, 2013.

Lu, Zhengdong, and Hang Li. A Deep Architecture for Matching Short Texts. Advances in Neural Information Processing Systems. 2013.

Ji, Zongcheng, Zhengdong Lu, and Hang Li. An Information Retrieval Approach to Short Text Conversation. arXiv preprint arXiv:1408.6988(2014).

Thanks

Q&A