A Semantic Level Matching Method for Unstructured Documents

Liu Hai, Chen Xiaoming

Guangdong Polytechnic of Science and Technology, Guangdong, China

Keywords: Semantic matching, Unstructured, Multi granularity

Abstract: This paper proposes a text semantic matching model, designs multiple features for text matching from word level, phrase level, sentence level and semantic level, and uses ranking learning method to fuse features. Experiments are carried out to verify the performance of the proposed method in text matching and real scene human-computer dialogue tasks. Experiments show that the multi granularity text semantic matching model can achieve high accuracy in text matching task, and has good domain migration ability.

1. Introduction

The expression of natural language generally has diversity and ambiguity. The same word may have different semantics in different contexts, and the same semantics can also be expressed by different words. The method based on word surface matching has poor effect in sentence level semantic matching, and is difficult to be applied in practice. With the development of deep learning technology, the method based on convolutional neural network has been proved to be effective in text matching task because it can map the text pairs to vector semantic space. On this basis, the researchers also proposed the use of attention mechanism to enhance the matching effect. Researchers have found that natural language sentences can be mapped to vector space through word embedding and neural network. Experiments show that this representation learning method based on neural network can go beyond word granularity and phrase granularity, and effectively improve the accuracy of sentence semantic similarity or relevance judgment. This paper first introduces the convolutional neural network text matching model, and designs two sentence level text matching features based on the model: one is the feature Fscr to measure the causal relationship between sentences, the other is the feature fsdr to measure the context relationship between sentences.

2. Distributed Convolutional Neural Network Based on Attention Mechanism

The model uses distributed convolution neural network to represent two sentences to be matched into two vectors of fixed length through neural network, and then uses the cosine distance of the two vectors in space to represent the semantic relationship between sentences. We use attention mechanism to express the original sentence vector expressions s x and sY is reweighted and mapped to the new vector expressions s'x and s'y. through this attention mechanism, words of different importance are given different weights to reflect sentence level semantics. Next, we will introduce the model step by step

(1) Input layer

The input layer of the model first maps the input statement pairs SX and sy into two word embedding matrices SX = [ex, 1,. . .]Ex, NX] and sy = [ey, 1Where ex, I ∈ rk and ey, J ∈ rk represent the word embedding corresponding to I in SX and j in SY respectively. Each word embedding is represented by a k-dimensional vector. According to SX and sy, we calculate the word similarity matrix m according to the following formula:

$$Mi, j=con (ex, i, ey, j) \quad (1)$$

Then the row attention vector VC and column attention vector VR are calculated respectively

$$Vc=max1\leq i\leq n\{M (i, \cdot) \}$$

$$Vr=max1\leq j\leq m\{m (\cdot, j) \} \quad (2)$$

  DOI: 10.25236/iccem.2021.044

Then, the distribution of attention weights α X and α y is calculated according to VC and VR

αx, k=exp (Vc (k)) /∑i=1 n exp (Vc (i))

αy, k=exp (Vr (k)) /∑j=1 m exp (Vr (j)) (3)

Finally, according to the distribution of attention weight, the s ′ X and s ′ y expressions of sentences are updated

S′ x=[e′ x, 1, …, e′ x, nx]=[αx, 1·ex, 1, …, αx, nx·ex, nx]

S′ y=[e′ y, 1, …, e′ y, ny]=[αy, 1·ey, 1, …, αy, ny·ey, ny] (4)

(2) Convolution layer

The convolution kernel WC ∈ RHK with window size h is used to calculate the h-gram feature of s ′ X. the object e'x, j: j + H-1 = [e'x, J, e'x, j + 1, e'x, j + H-1] is a matrix formed by embedding and splicing continuous h words of J,

ci, j=tanh (wc·e′ x, j:j+h−1+b) (5)

The j-th characteristic Ci, J with convolution kernel W and variance B as parameters are obtained. An h-size window uses the i-th convolution kernel WC, I to perform the above operation, and then we can get the convolution vector CI ∈ rn-h + 1, CI = [CI, 1, CIWe use n convolution kernels to perform convolution operations and get C1, C2C n is calculated in the same way, using other n convolution kernels to calculate n convolution vectors of s ′ y in another sentence, which represent CN + 1, CN + 2C2n.

(3) Pooling layer

Firstly, the 2n local feature vectors obtained by convolution operation are pooled maximally: ˆ CI = max {CI}; then, the first n elements constitute the new expression CX of the sentence SX, and the last n elements constitute the new expression cy of the sentence sy, and the vectors CX and CY retain the global features of the sentence.

(4) Output layer

The output of the nonlinear transformation function is used to convert CX into two vectors VX as the final vector expression of two sentences, VX = tanh (Wo · Cx), where Wo is the parameter. For CY, we use a similar operation to get another sentence for vy. The purpose of each parameter in the above model is to minimize the maximum interval loss function, which is trained by the random gradient descent method

L=max{0, M-con (Vx, Vy+) +con (Vx, Vy-) } (6)

Where vy + and vy - are the corresponding expressions of positive samples and negative samples respectively, and M is the constant representing the maximum interval.

Sentence level feature design based on convolution neural network model

Based on the above model, a feature Fscr is designed to measure the causality between sentences

Fscr (Q, s) =con (CNNscrQ (Q) , CNNscrs (s)) (7)

Cnnscrq (·) and cnnscrs (·) represent two functions that map natural language sentence input to a fixed length vector. The vector expression of two sentences is obtained, which measures the relationship between them by calculating the cosine distance between them. Cnnscrq (·) and cnnscrs (·) use supervised question and answer pairs as the corpus of model training, and match the first sentence (spre) and second sentence (snext) of the current sentence with the computational features of question matching. In addition, a feature fsdr is designed to measure the context relevance between sentences,

Fsdr (Q, s) =con (CNNsdrQ (Q) , CNNsdrs (s)) (8)

Cnnsdrq (·) and cnnsdrs (·) are two sub networks of distributed neural networks, which use supervised sentence pairs as the corpus of model training. The first sentence and the second sentence of the current sentence are also associated with the matching problem of computational context sensitive features fsdrpre - and fsdrnext, respectively.

3Semantic features

It is very important to measure the semantic matching between the answer candidates and the user's conversation. When people are in conversation, they should not only consider the

context of the conversation, but also select words and sentences according to the context of the conversation and the speaker's domain knowledge and other semantic information. Therefore, semantic level text matching features are designed to construct the conversation topic and domain knowledgeIn this part, we will introduce four semantic level matching features: fre based on entity relationship, FTE based on entity type, fstm based on supervised topic prediction and futm based on unsupervised body protection model.

3. 1 Feature Extraction Based on Knowledge Map

Firstly, two semantic level text matching features (fre and FTE) based on knowledge map are introduced, and a community knowledge map project named freebase is established in 2007. Community members can edit knowledge according to the pre-defined architecture. If entities in the knowledge base are regarded as nodes in graph theory, the relationship between entities is regarded as the boundary between nodes, and then freebase knowledge base is constructedA graph based network structure will be formed. Freebase describes the knowledge of more than 20 million entities. Entity relationship feature (FRE) and entity category feature (FTE) are designed to measure the semantic relationship between problems and nodes. A: in freebase knowledge base, a knowledge can be described by a triple (<esbj, rel, eobj>). In the triple, esbj is an entity, representing the subject of knowledge; eobj is also an entity, representing the object of knowledge; rel represents the relationship between the subject and the object of knowledge. For example, we can use “mabobo, contribution, e-commerce” to express the knowledge that “mabobo contributes to e-commerce”. Freebase also defines a special relationship, that is, the relationship between an entity and its attribute type. For example, mabobo, type, name describes “mabobo” as a person's name.

For some practical natural language questions, a natural language question Q and its corresponding answer a can be parsed and mapped to one or more knowledge fragments in the knowledge base. Assuming that only the question is mapped to a single knowledge, we can analyze the entity related to the question and the entity relationship involved in the question from question Q, such as “where was Ma Yun bornWhat's in itYou can map to the knowledge of “Mayun, place of birth, Hangzhou” and return “Hangzhou” as the user's answer. Based on the mapping relationship between natural language and knowledge, we can establish the relationship between entity relationship and its natural language expression, as well as the relationship between entity attribute and its natural language expression. For example, in the previous example, the natural language expression “where Ma Yun was born” and the entity relationship place of birth have a mapping relationship with the entity attribute “place”, which means that the question is about the “birthplace” relationship between two entities, and the answer should be the entity with the “birthplace” attribute. An example is given to illustrate how to establish the relationship between natural language and entity and the entity attribute through freebaseWe can see that knowledge triples can be used as a bridge between natural language expressions and their relationship semantics and attribute semantics. We design relevant features to describe the relationship semantics (FRE) and attribute semantics (FTE) contained in natural language.

Table 1 Mapping Relationship between Knowledge and Natural Language Problems in Freebase

1	Question Knowledge triples (< esbj, rel, eobj >) Knowledge object and attribute (< eobj, type >)	What do you eat on Mid-Autumn Festival? Eating zongzi in Mid-Autumn Festival Zongzi, food
2	Question Knowledge triples (< esbj, rel, eobj >) Knowledge object and attribute (< eobj, type >)	When is the Mid-Autumn Festival? August 15 is the Mid-Autumn Festival 8May 15, date

$$fSTM(Q, S) = con(Ystm(Q), Ystm(S)) \quad (9)$$

Ystm is the model of CDSSM after training.

Example 1 question: what does Jim neutral do? Knowledge triple (< esbj, rel, eobj >) < Jim neutral, functional character occupation, inventor > knowledge object and attribute (< eobj, type >) <

inventor, functionaluniverse. character Which forest is fires Creek in? Knowledge triplet (<esbj, rel,Eobj >) < res Creek, contained by, Nantahala National Forest > knowledge object and attribute (< eobj, type >) < Nantahala national forest, location. location>

Based on the set of problem knowledge triples, we can get the set of knowledge relation semantic pair (< question, rel >) and knowledge attribute semantic pair (< question, type >). For freebase and other manually organized knowledge bases, the relationship between entities and the attributes of entities are closed sets, denoted as t. We define that for a given natural language user conversation Q and a semantic tag candidate tag, the conditional probability of their correlation is as follows:

$$P(tag|Q) = \exp(\text{cosine}(y(tag), y(Q))) / \sum_{tag \in T} \exp(\text{cosine}(y(tag), y(Q))) \quad (10)$$

Where y (·) is a function that maps natural language conversation or semantic markers to vector space. We use cdssm model as function y (·) to get the vector expression of user session and semantic tag respectively, and then define the feature fre and FTE as

$$fRE(Q,S) = \text{cosine}(yRE(Q), yRE(S))$$

$$fTE(Q,S) = \text{cosine}(yTE(Q), yTE(S)) \quad (11)$$

Using question relation pairs (< question, rel >) to train yre (q) and yre (s), using question type pairs (< question, type >) to train YTE (q) and YTE (s).

4. 2 Topic Based Features

For the dialogue scene, whether the topic of the context is consistent or not is another important semantic information. The more suitable reply candidates should have stronger similarity with the user conversation at the topic level. We design two topic information features to judge the topic consistency of user conversation and reply candidate sentences. Among them, one is judged by the consistency of topic distribution, and the other is judged by the result of topic prediction. The former obtains topic distribution based on unsupervised method, so it is called unsupervised topic feature futm; the latter obtains topic distribution based on supervised method, so it is called supervised topic feature fstm. The unsupervised topic feature FTM calculates the average cosine distance between the topic vector of each non stop word in user conversation Q and the topic vector of each non stop word in candidate sentence s

$$fUTM(Q,S) = |Q| \sum i=1 \ln(\sum |S| \, j=1 \, \text{cosine}(vQi, vS\, j) \, |S|) / |Q| \quad (12)$$

VW = [P (t1|w),. . . , P (tn|w)] t represents the topic vector of the word, and P (ti|w) represents the probability that the word w belongs to the topic ti. After the attribute corpus is given, the topic distribution P (ti|w) of words in the corpus can be estimated by topic models such as plsi (probabilistic Latin semantic indexing) or LDA (Latin Dirichlet allocation). The unsupervised topic distribution method has two disadvantages: one is that the number of topics is usually given in advance as a super parameter, which is difficult to adapt to the needs of customizable corpus; the other is that the topic distribution estimated by topic model is lack of interpretability. In this paper, we propose a supervised topic feature fstm based on topic prediction. The cdssm algorithm is used to train the topic expression function of sentences on the sentence topic pair, which is the same as training the semantic expression function based on knowledge. We crawled a large number of sentence topic pairs from Wikipedia as training data. The topic is the name of each chapter in Wikipedia, such as “history”, “geography”, “early career”, etc. , and the first sentence under the chapter is the sentence matching the topic. Wikipedia is a manual edited encyclopedia. Usually, the title of a chapter can summarize the topics of the following articles. The first sentence of each chapter often contains the most information of this chapter. For example, “Dunhuang is located at the northern edge of the Qinghai Tibet Plateau and the western end of the Hexi corridor. “It's the first sentence in the chapter “geography” on the “Dunhuang city” page of Wikipedia. We will combine this sentence with “geography” to form a sentence topic example. Based on this, supervised topic features can be defined as

$$fSTM(Q,S) = \text{cosine}(ySTM(Q), ySTM(S)) \quad (13)$$

Ystm is the cdssm model after training.

5. Experiment

All three word level features FWM, fw2w and fw2v are represented by fword. Among them, FWM does not need to be trained in advance. For the word level translation feature fw2w, we use the 11. 6 million "problem similarity problem" sentence pairs published by fader et al. To estimate the translation probability between words. For the word vector feature fw2v, the word2vec model is trained on the English Wikipedia corpus, and the length of the word vector is set to 300 dimensions. We use fphr to represent two phrase level features FPP and FPT. For phrase rewriting feature FPP, we use 500000 Chinese English Parallel Corpus to extract phrase translation table, and then obtain the rewriting probability of phrases in the same language. The parallel corpus mainly comes from the following data sets: ldc2003e07, ldc2003e14, ldc2005t06, ldc2005t10, ldc2005e83, ldc2006e26, ldc2006e34, ldc2006e85 and ldc2006e92. For the phrase translation feature FPT, first of all from the English community Q & a website Yahoo answer!Four million pairs of "question answer" sentences have been crawled up. The answers in community Q & A are often long, so only the first sentence is selected to construct the "question answer" corpus. In contrast, the first sentence in the answer usually contains the most information and can summarize the whole answer better than other sentences. Based on the "question answer" sentence pair, the phrase translation table is constructed and the phrase translation probability is obtained. Fsent is used to represent six sentence level features: Fscr pre, Fscr, Fscr next, fsdr pre, fsdr and fsdr next. For the causal association feature Fscr, we use the 4 million "question answer" sentences mentioned above to train the convolutional neural network based on attention mechanism. For the context sensitive feature fsdr, 500000 "upper sentence - lower sentence" pairs are randomly selected from Wikipedia articles to train the model. FSEM is used to represent four semantic features: fre, FTE, futm and fstm. We build the "question relation" sentence pairs for training fre and the "question type" sentence pairs for training FTE based on the simple questions data set. The 108422 questions in the simple questions dataset were written by the data taggers according to the knowledge triples in the freebase knowledge base. According to the "problem knowledge" pair, the corresponding "problem relationship" pair and "problem type" pair can be generated according to the construction method in Section 3. 2. For the unsupervised topic model, lightlda algorithm is used to calculate the distribution of topic words in all English Wikipedia sentences, and the number of topics is set to 1000. For the supervised topic model, we construct a corpus of 25000 topics and 4 million sentence topic pairs based on unstructured document retrieval technology. For the Chinese experiment, 17 million pairs of "question similar question" and 5 million pairs of "question answer" sentences were crawled from Baidu know. In the same way as in the English experimental setting, only the first sentence is retained in the answer part of the "question answer" sentence pair. The problem similarity problem corpus is used to train Chinese word level translation feature fw2w. The question answer corpus is used to train the phrase level translation model FPT and the sentence level causal association model Fscr. The Chinese English bilingual alignment corpus used in the phrase rewriting model FPP is the same as that used in the English experimental setup. Here, only the source language and the target language are interchanged to generate the probability table of phrase substitution in Chinese. Chinese Wikipedia is used to train Chinese words to embed fw2v, estimate topic distribution and get futm, construct "upper sentence lower sentence" corpus and train context related feature fsdr. We constructed 1. 3 million "sentence topic" pairs on Chinese Wikipedia and trained fstm. In view of the fact that there is no open knowledge base in Chinese, we ignore the two features of knowledge base, fre and FTE.

This paper reports the experimental results of the proposed matching model on English wikiqa data set and Chinese dbqa data set, compares it with other published methods, and discusses the corresponding experimental results.

Table 2 Comparison of Matching Models

	method	MAP	MRR
1	LCLR	59. 93%	60. 68%
2	BiCNN	65. 20%	66. 52%
3	ABCNN	69. 21%	71. 08%

4	DocChat	68. 25%	70. 73%
5	DocChat+BiCNN	70. 08%	72. 22%

The table reports the performance of the matching model (represented by docchat) and other baseline models on the English wikiqa dataset. Consistent with the setting of baseline technology on wikiqa data set, all samples with wrong or correct candidates are excluded in this experiment. The first four rows in the table show the performance of the four baseline methods on the wikiqa dataset. They are: (1) LCLR method integrates a large number of manually customized semantic features; (2) bicnn uses a convolution neural network architecture to convolute all two adjacent word vectors, and finally average pool to get sentence expression; (3) NASM uses long-term and short-term memory based on potential random attention mechanism (4) abcnn is a convolutional neural network model based on attention mechanism, which performs well in many matching types of natural language processing tasks. The performance of the above four baseline methods on wikiqa data sets are the results reported in their respective papers. It can be seen that docchat can achieve results close to the baseline method. It is worth noting that each independent feature used in docchat's multi granularity semantic matching model is not trained with wikiqa's training set, only the wikiqa's development set is used to adjust the weight in feature fusion. This relatively independent attribute of data sets can bring good adaptability to docchat technology, make it easier for docchat technology to be used in different data sets, and provide the basis for docchat technology to serve customizable scenarios. It can be seen from line (6) that when other models trained with wikiqa training set are integrated, the method can achieve better results.

References

[1] Bordes A, Boureau Y L, Weston J. Learning End-to-End Goal-Oriented Dialog[A]. International Conference on Learning Representations (ICLR)[C], 2017

[2] Watkins D. Global Smart Speaker Vendor & OS Shipment and Installed Base Market Share by Region: Q4 2017[EB/OL]. https://www. strategyanalytics. com/, 2018.

[3] Al-Rfou R, Pickett M, Snaider J, et al. Conversational contextual cues: The case of personalization and history for response ranking[J]. arXiv preprint arXiv:1606. 00372, 2016

[4] Zhao T, Eskenazi M. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning[A]. 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue[C], 2016. 1.

[5] Dong L, Wei F, Zhou M, et al. Question answering over freebase with multi-column convolutional neural networks[A]. Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACLIJCNLP)[C], 2015. 260–269

[6] Berant J, Liang P. Semantic parsing via paraphrasing[A]. Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics (ACL)[C], 2014. 1415–1425

[7] Song Y, Yan R, Li X, et al. Two are Better than One: An Ensemble of Retrieval-and Generation-Based Dialog Systems[J]. arXiv preprint arXiv:1610. 07149, 2016. .

[8] Qiu M, Li F L, Wang S, et al. Alime chat: A sequence to sequence and rerank based chatbot engine[A]. Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (ACL)[C], 2017. 498–503

[9] Yan R, Song Y, Wu H. Learning to respond with deep neural networks for retrieval-based humancomputer conversation system[A]. Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval[C]. ACM, 2016. 55–64.

[10] Tan M, Santos C N, Xiang B, et al. Improved Representation Learning for Question Answer Matching[A]. Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)[C], 201