A Deep Top-K Relevance Matching Model for Ad-hoc Retrieval(DTMM)

Zhou Yang[1], QingFeng Lan[2], Jiafeng Guo[3], XiaoFei Zhu[1], YanYan Lan[3], YiXing Fan[3], Yue Wang[1] and Xueqi Cheng[3]

[1]School of Computer Science and Engineering, Chongqing University of Technology

[2]University of Chinese Academy of Sciences

[3]CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190,China

CONTENTS

PART 1

Motivation

mismatch problem awalys exits in traditional models and neural information retrieval models

for example

Query: A little dog was running happily on the road.

The model is judged to be irrelevant.

Document: A puppy runs cheerfully on the path

- One of the important issues in general information retrieval is vocabulary mismatch.

Methods

1.Query expansion

puppy cheerfully path

Query: A little dog was running happily on the road.

The model is judged to be relevant.

Document: A puppy runs cheerfully on the path

Query expansion is the standard technique for reducing vocabulary mismatch

2.Query expansion

Query: A little dog was running happily on the road.

The model is judged to be relevant.

Document: A puppy runs cheerfully on the path

Dog happily road

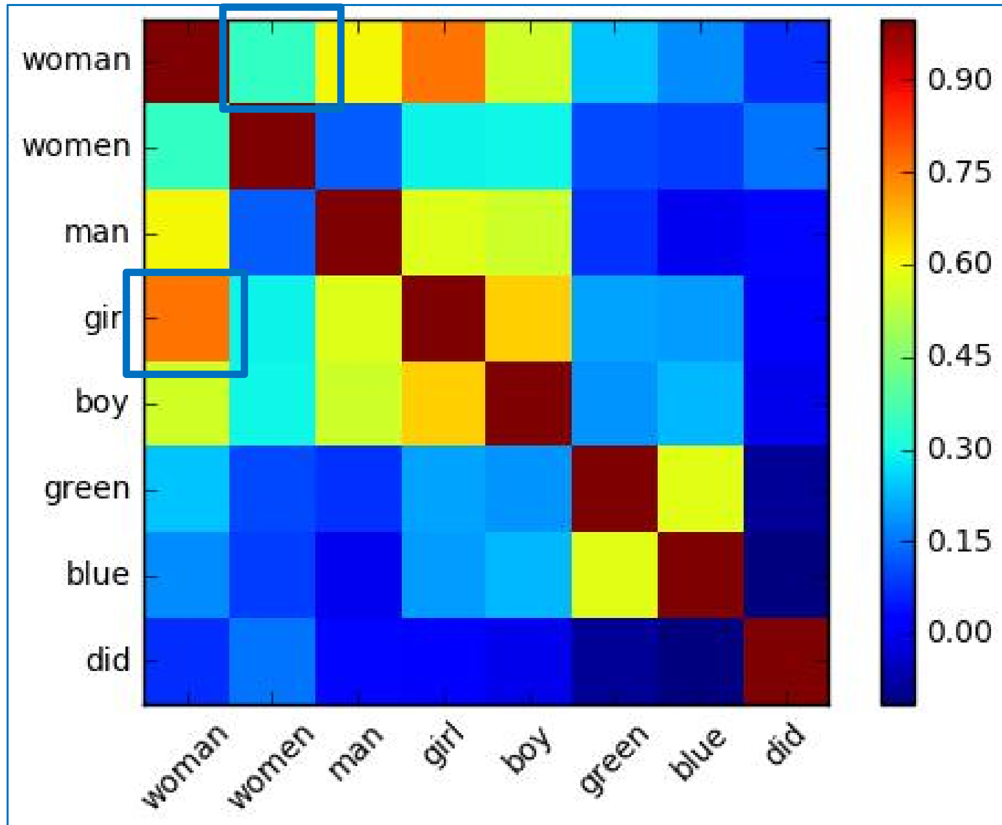
A different approach would be to expand the documents by adding related terms.

PART 2 Observation

Woman: girl > man > boy > women > green > blue > did

Women: woman > boy > girl > did > man > green > blue

```
woman woman 1.0
woman women 0.3451595268
woman man 0.607956254336
woman girl 0.761190251497
woman boy 0.558522930154
woman green 0.24118403927
woman blue 0.178044251325
woman did 0.0751838683173
women woman 0.3451595268
women women 1.0
women man 0.126646555737
women girl 0.292825346454
women boy 0.298552943639
women green 0.104096393379
women blue 0.0930137564485
women did 0.152766770859
```



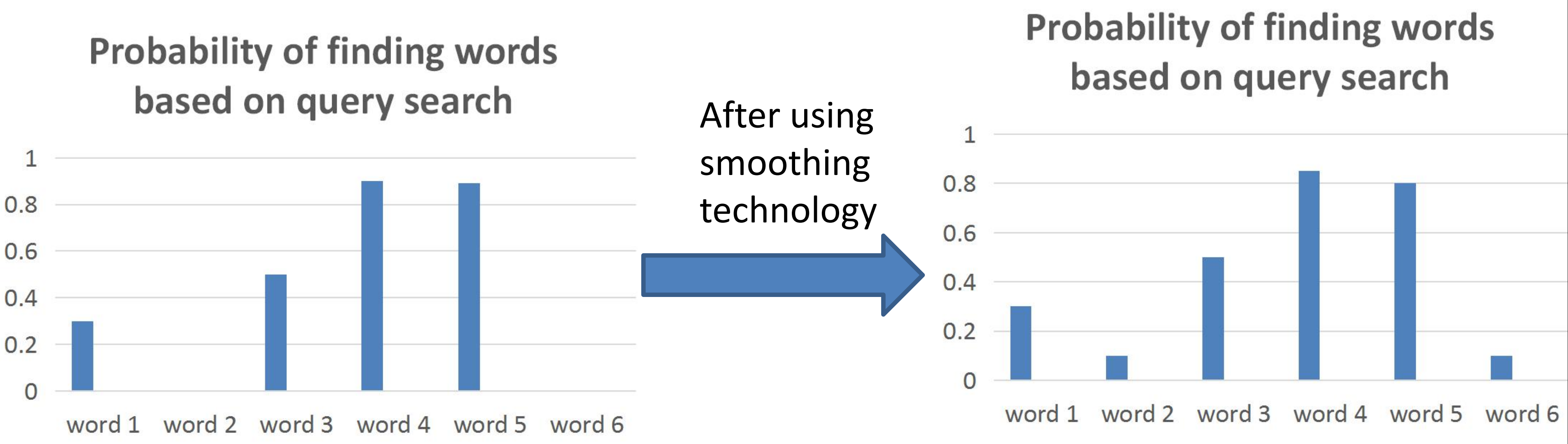
How to solve this problem?

The similarity between Woman and girl higher than it between woman and women.

Even more unreasonable is that the similarity between *women* and *did* higher than *women* and *man*'s.

smoothing technology for neural information retrieval models ?

- For documents represented as language models, this is equivalent to smoothing the probabilities in the language model so that words that did not occur in the text have non-zero probabilities. [Croft et al, 2010]



- The core idea of smoothing technology is to “rob the rich and help the poor”, mainly to solve the problem of data sparsity.

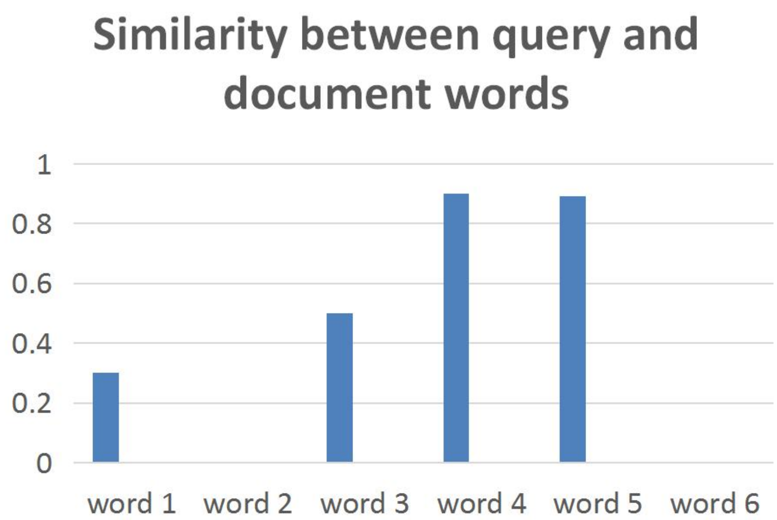
PART 3

Method

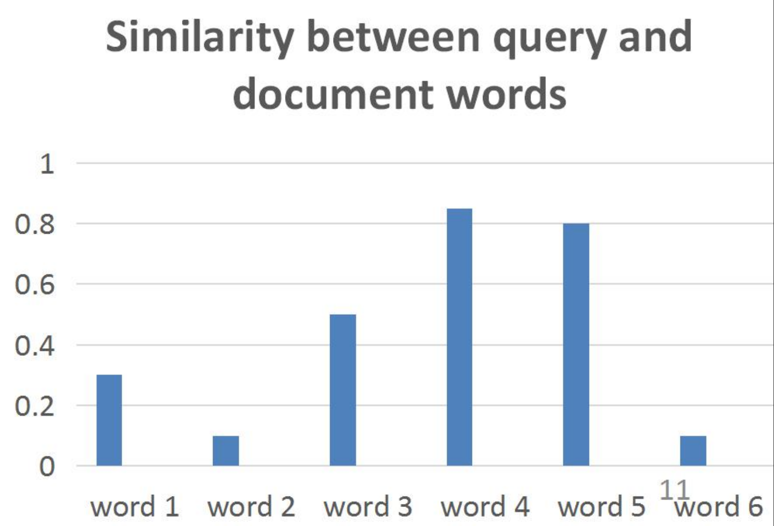
Deep Top-K Relevance Matching Model(DTMM)

Assumptions

Adding the weight of each document word to the similarity between the query word and the document word to compensate for the unreasonable similarity.



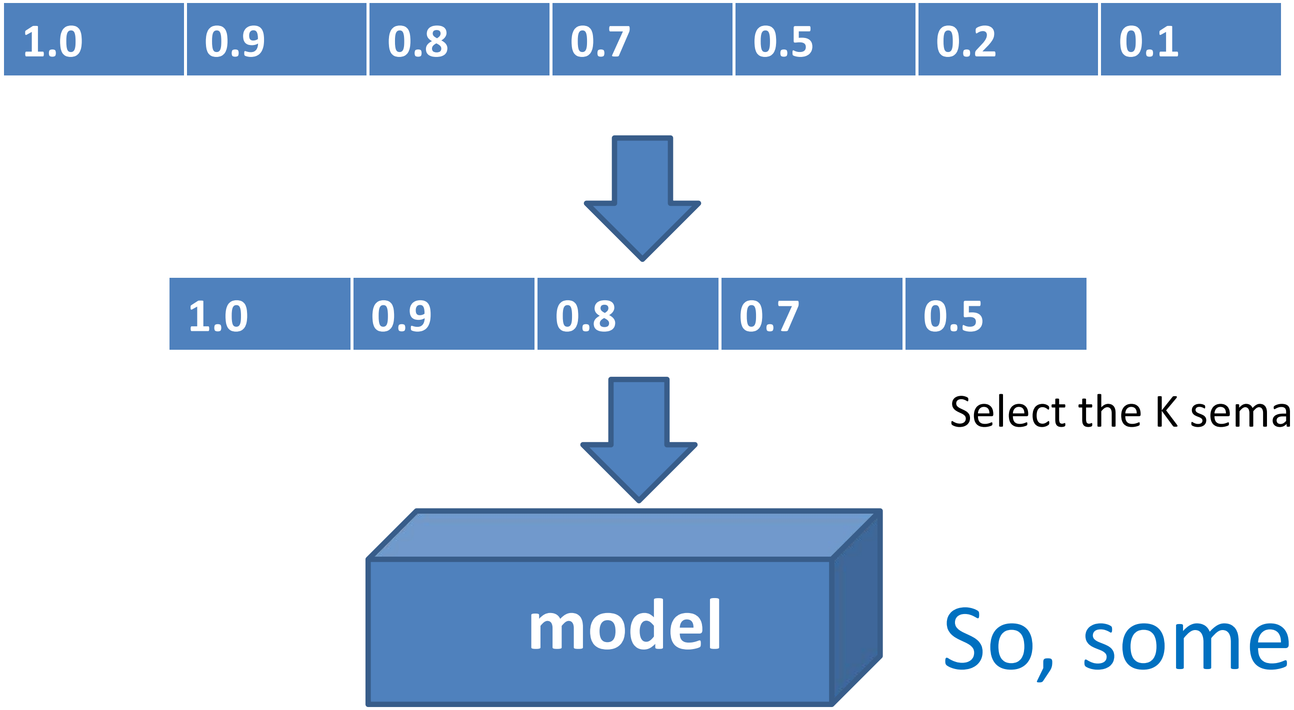
Adding weight of document word



Assumptions

It is not enough to fill the deviation. We should also remove the noise introduced after filling the deviation.

Similarity between a query word and document words



Select the K semaphore input model.

So, some questions!

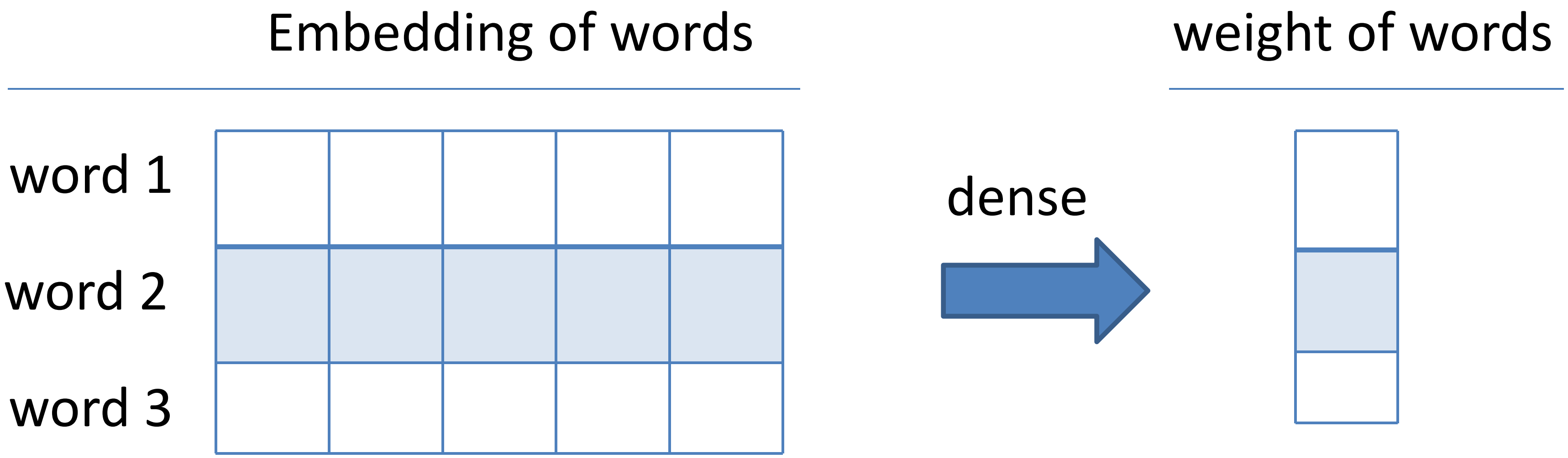
Question 1

- How to calculate the weight of all words in query and document? Should we use idf to calculate weights like BM25?

$$\text{IDF} = \log\frac{N}{n}$$

N represents the total number of documents in the dataset. n indicates the number of documents containing the word.

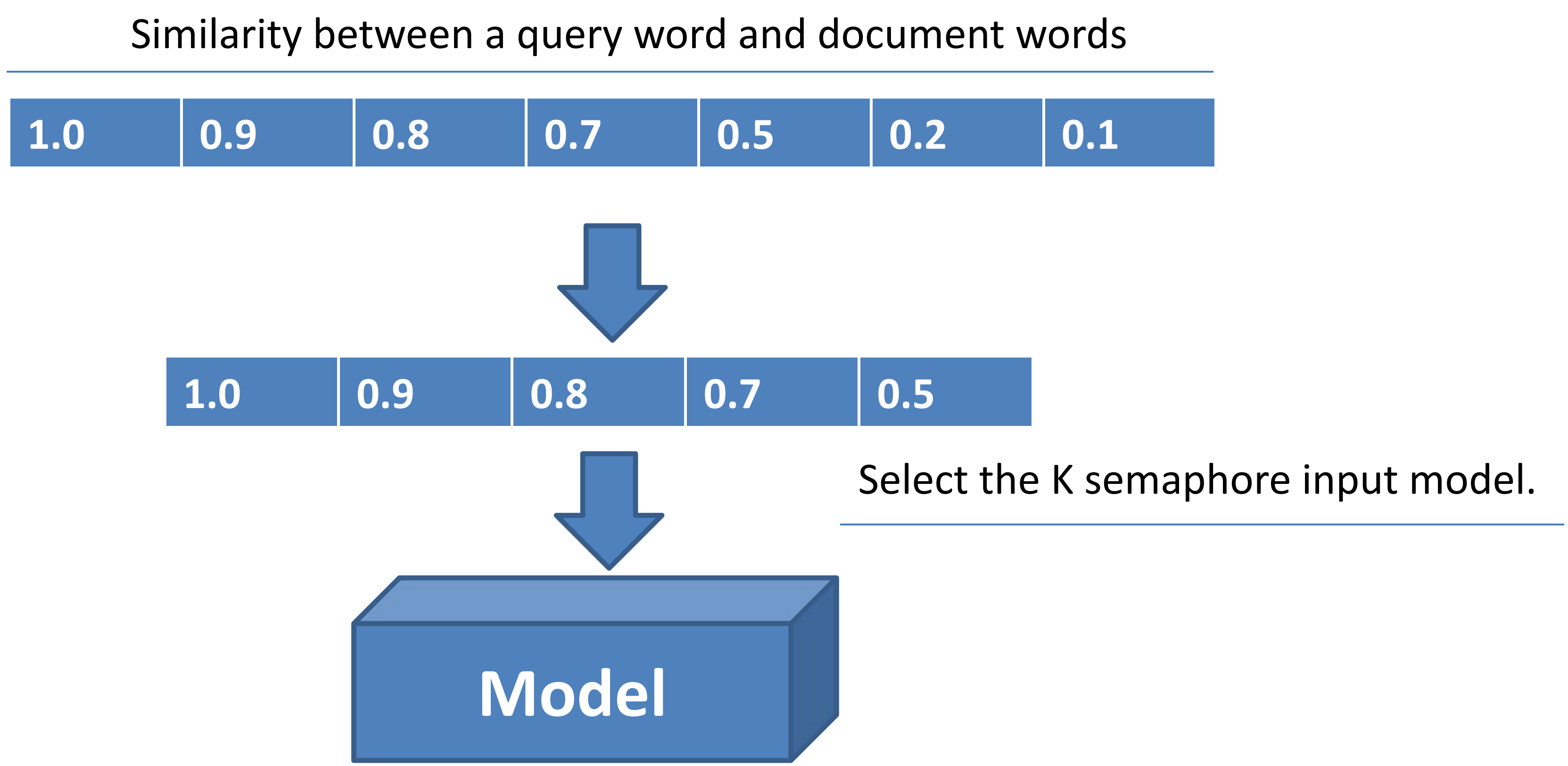
Our conclusion



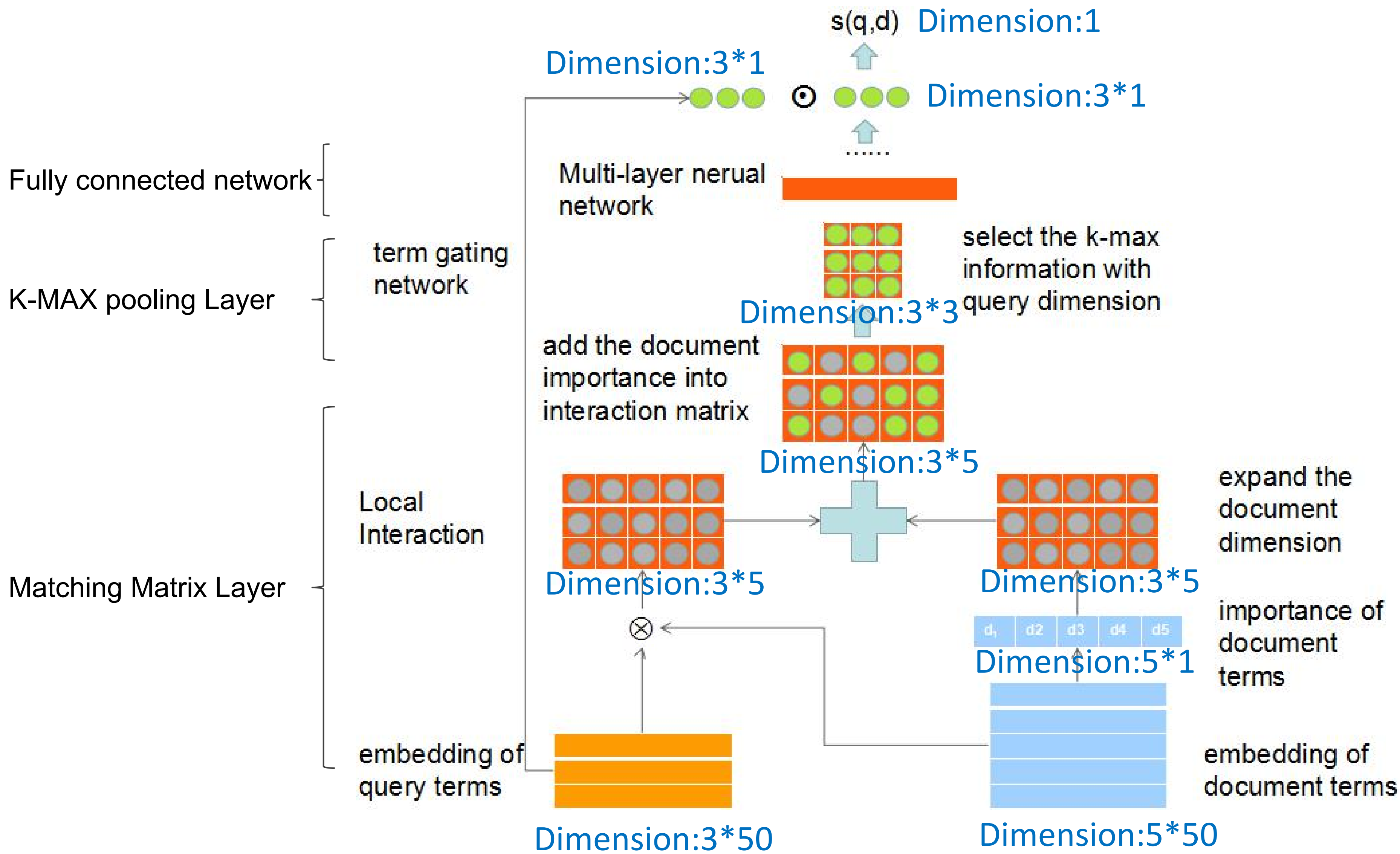
Question 2

➢ How do we determine the threshold for singals?

Considering that the number of features of different data sets is different, we set the hyperparameter k and take the top k strongest semaphores.



Models construct



- Embedding :Trained by glove model.
- Embedding size: 50 dimension size.
- K-max pooling layer size:In mq2007, the robust04 data set is set to 512, 15 respectively.
- Multilayer neural network size: The size of the multi-layer neural net-
- work is set to [512,512,256,128,64,32,16,1] with mq2007 dataset, while set to [15,10,1] with robust04.
- Model optimization: Optimization using Adam optimizer, with e = 1-5,learning rate = 0.001 and batch size =100.

Loss function

$$L(\theta) = mean[\sum_{q} \sum_{d^+ \in D_q^+, d^- \in D_q^-} max(0, 1 - s(q, d^+) + s(q, d^-))]$$

- In detail, θ represents all the parameters to be learned in the model, q denotes query, d+ comes from the positive sample document sets D+ , which represents the documents that is positively related to the query. comes from the negative sample document sets D− , which represents the documents that is not related to the query.

PART 4

Experimental Analysis

Dataset

- Million Query Track 2007: It is called MQ2007 for short. The data set is a subset of the LETOR4.0

- robust04:The topics are collected from TREC Robust Track 2004.
- Here the Robust04-Title means that the title of the topic are used as query.

Table1.Statistics of collections used in this study .Here we tested our model DTMM on two data sets MQ2007 and robust04.

	MQ2007	robust04
query number	1501	250
document number	58730	324541

Performance Metrics

precision

$$\text{Precision}=\frac{retrieved \cap relevant}{retrieved}$$

- The meaning of prescision is the proportion of related documents retrieved by the model to the retrieved documents.

the mean of average precision scores(MAP)

$$AveP = \frac{1}{R} \times \sum_{r=1}^{R} \frac{r}{position(r)}$$

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

- AvgP is computed by dividing the first relevant document by its position in the sorting result, and MAP is the average of multiple query results

Performance Metrics

Normalize Discounted cumulative gain(NDCG)

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

- Suppose each position is sorted from small to large, and their value is decremented. For example, you can assume that the value of the i-th position is $\frac{1}{log_2\,(i+1)}$
- IDCG is the DCG in the ideal case, that is, the maximum value of DCG for a query statement and p.

Ranking accuracy

Table 2. Comparison of different retrieval models over the MQ2007.

Model	NDCG@1	NDCG@3	NDCG@5	NDCG@10	P@1	P@3	P@5	P@10	MAP
BM25	0.358	0.372	0.384	0.414	0.427	0.404	0.388	0.366	0.450
DSSM	0.290	0.319	0.335	0.371	0.345	0.359	0.359	0.352	0.409
CDSSM	0.288	0.288	0.297	0.325	0.333	0.309	0.301	0.291	0.364
ARC-I	0.310	0.334	0.348	0.386	0.376	0.377	0.370	0.364	0.417
DRMM	0.380	0.396	0.408	0.440	0.450	0.430	0.417	0.388	0.467
ARC-II	0.317	0.338	0.354	0.390	0.379	0.378	0.377	0.366	0.421
MatchPyramid	0.362	0.364	0.379	0.409	0.428	0.404	0.397	0.371	0.434
DTMM	0.458	0.459	0.468	0.499	0.517	0.479	0.458	0.426	0.504

- The improvement of DTMM against the best deep learning baseline (i.e. DRMM) on MQ2007 is 20.6% wrt NDCG@1, 15% wrt P@1, 8% wrt MAP, which illustrates the superiority of our model on the IR task.

Table 3. Comparison of different retrieval models over the robust04.

Model	NDCG20	P@20	MAP
BM25	0.418	0.370	0.255
DSSM	0.201	0.171	0.095
CDSSM	0.146	0.125	0.067
ARC-I	0.066	0.065	0.041
DRMM	0.431	0.382	0.279
ARC-II	0.147	0.128	0.067
MatchPyramid	0.330	0.290	0.189
DTMM	0.463	0.432	0.314

- On this data set, DTMM also achieves the best effect, compared to the best model DRMM. the improvement of DTMM against the best deep learning baseline (i.e. DRMM) on robust04 is 7.4\% wrt NDCG@20, 13\% wrt P@20, 12.5\% wrt MAP, respectively.

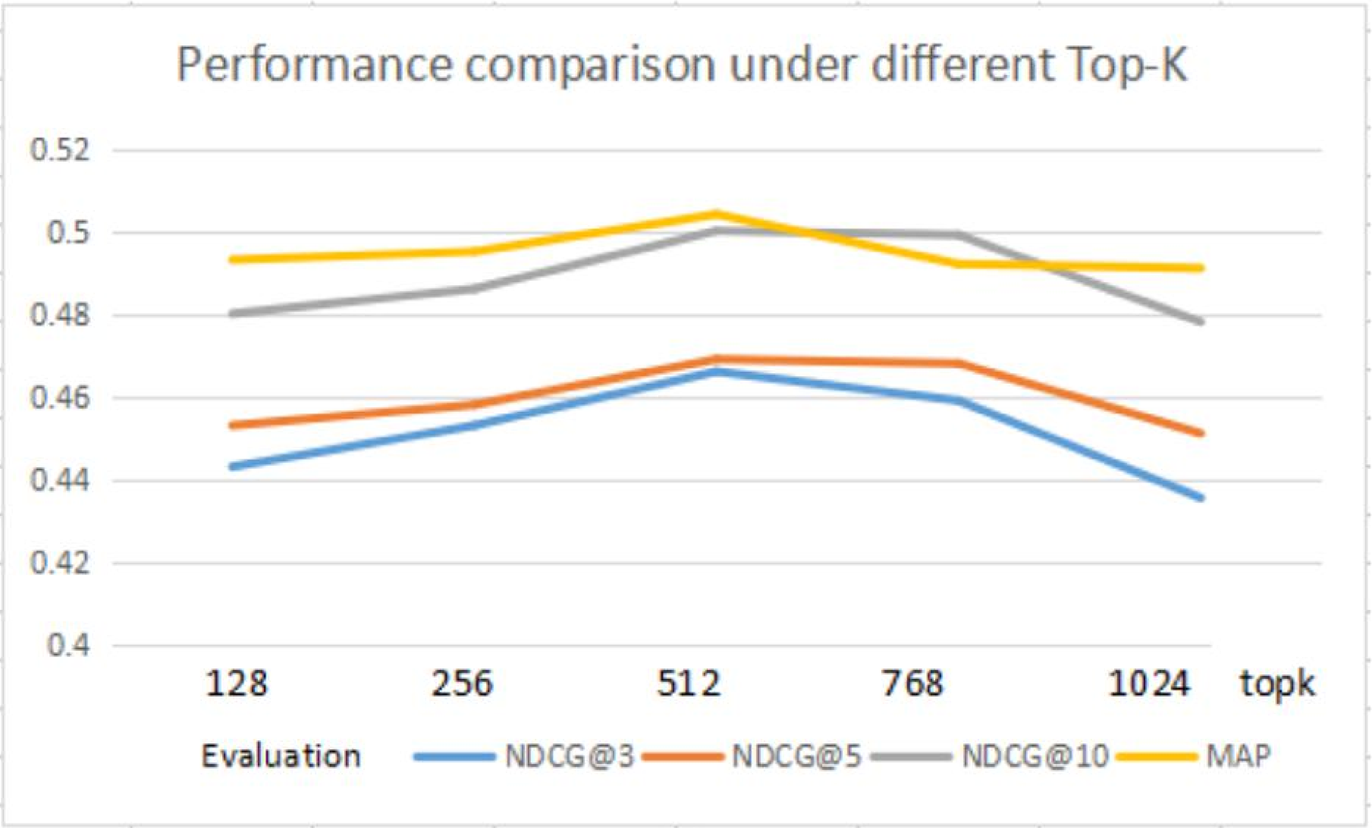
The influce of document words importance

Table 4. Comparison of different version of DTMM. Where $DTMM_{no}$ represents the model without document words importance, the other is the complete model.

Model	NDCG@3	NDCG@5	NDCG@10	MAP
$DTMM_{no}$	0.424	0.435	0.469	0.490
DTMM	0.459	0.468	0.499	0.504

- DTMM was higher than the incomplete model **8.25%, 7.58%, 6.39%, 2.85%** respectively.

Performance on different k-max pooling layer of DTMM



- Obviously, with the parameter selection from small to large, the performance of the model first improves and then decreases.

PART 5

Conclusion

Summary of Contributions

1. We propose to use the weight of the document word to compensate for the deviation of the similarity between words and words by embeding.

2. In the text matching process, not all words are important. And it is necessary to filter out unimportant words.

Future Works

1. The interaction matrix information should be richer, rather than simply constructing with similarity

2. We will further alleviate the mismatch problem from the perspective of multi granularity.

We will continue our research from the above two aspects.

References

- ✓ 1. W. B. Croft, D. Metzler, and T. Strohman. Search engines: Information retrieval in practice, volume 283. Addison-Wesley Reading, 2010.
- ✓ 2. J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A deep relevance matching model for ad-hoc retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pages 55–64. ACM, 2016.
- ✓ 3. J. Guo, Y. Fan, Q. Ai, and W. B. Croft. Semantic matching by non-linear word transportation for information retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pages 701–710. ACM, 2016.
- ✓ 4. B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In Advances in neural information processing systems, pages 2042–2050, 2014.
- ✓ 5. P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pages 2333–2338. ACM, 2013.
- ✓ 6. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
- ✓ 7. T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746–751, 2013.

THANGK YOU FOR LISTENING