A pipline solution based text classification for product entity query

[1]*Si Sun,* [2]*Chenyu Jin,* [1]*Xiaofeng Chen,* [1]*FengMing Cao*

[1]Ping An Group, [2]China Literature Limited

{ sunsi537, chengxiaofeng061, caofengming777}@pingan.com, jinchenyu@yuewen.com

Abstract

In e-commerce, the information of commodity uploaded by the seller often needs to be linked to the standard entity in the standard entity library to facilitate the provision of search service and commendation service for users. We propose a pipeline solution with 3 modules to search the standard entity by the title text of a commodity from a large standard library, including filter module, rough ranking module and exact ranking module. We also innovatively propose a method of reranking hard sample based the three modules. In our method. We first filter out the valid entities from the entity library by base rules in module 1, then a multiclass classification model and a following binary classification model are trained respectively. In module 2, the multiclass classification model selects entities with the top N highest predicted probability as candidate entities of module 3, and the candidate entities will be rearranged by binary classification model in module 3. Lastly, for the hard sample whose highest probability of belonging to a entity still belows to a threshold in module 2, we expand their candidate entities to all valid entities to rerank it again in module 3, where the entity ranked 1st in the module 3 will be choosed the final matching entity. At the same time, we test our solution on the dataset of CCKS2020 Product Entity Query provided by Alibaba. The results show that our representation achieves the superior performance on the dataset.

key words: Entity Query， Entity Match， Text Classification

1. Introduction

Product entity query based text title aims to quickly match the entity in the massive stardand entity library by the deep understanding the text title in semantics. This task is different from regular text query whose query text is generally more detailed with key words to accurately obtain search results while the desired result is uncertainty. For example, for a text query like "日本敏感肌卸妆水", the expected result may be anyone makeup remover for sensitive skin made in Japan. Oppositely, the query in title-based product entity query task is ambiguous while the result is accurate. The entity is clearly determined when the seller uploaded the product information to the e-commerce platform, but the title text information is usually incomplete and ambiguous for the limit of text length of title as well as the sellers want to get people's attention by exaggerated title. The information in the title is relatively one-sided just like the title of the product "一年卖出三千万，日本老品牌敏感肌卸妆膏，人手一只". If only based on limited title text information, it is difficult to for us to match the corresponding makeup remover. More necessary keypoint description about the product is often a comprehensive expression of multimodal data which is out of our reach in this situation. We can only get the text title information of a product. The difficulty of this task lies in the existence of extremely similar or confusing entities in the product library while the title text information is incomplete or cryptic, as well as there are not enough samples for some entity.

More recently, the general solution of text query firstly searches from the database which an inverted index has been created in the database for rough match to candidate results, and then uses a DNN model like DSSM[1] to rank the results. However, in our situation, when the title provided by the seller doesn’t express any key information, the result of matching by creating inverted index is poor.

In this work, we propose a pipeline solution with three modules of filter, rough ranking and exact ranking, we also innovatively propose a method of reranking hard sample based the three modules. Firstly, in filter module, we filter out the valid entities from the whole product library by some basic rules to reduce the number of the entity which will be used to match title texts in the following steps, then we train a multi-classification model with the entity as sample's class label. The entities with the top N highest predicted probability of a sample are choosed as candidate entities for the next module. In the following module, all text features of a entity concated with the title text is used for a detailed binary classification model to rerank the previous order, we choose the result with the highest probability as the matched entity. In the module 2, for the samples with the highest probability of being predicted to belong to an entity but still below the threshold we set, we consider them as hard samples. For these hard samples, we expand their candidate entities to all the valid entities and predicate again in the modules 3 to recorrect the result before.

We conducted the experiment on the dataset of CCKS2020 Product Entity Query. The dataset comes from Alibaba e-commerce platform which is created by users in real life. The dataset provides text title of product which seller created and a standard product library. Each product entity contains the product's name, factory, ingredients, efficacy, origin, specifications, etc. This task aims to matching each title with the normative entity in the product library, and it will facilitate to user search service or user recommendation service lately on the platform. The experiment results show that the accuracy of our solution on this data set achieve 0.87+, and it ranks second in the task.

2. Related work

Getting the entity to match user generate content from a massive standard entity library is the pre-order task for subsequent user search and product recommendation. It is also a routine task in the NLP field. Recently, model about natural language processing has become larger and more diverse. At the same time, the structure of existing models has evolved from shallow to deep. Conventional solution in text query includes retrieval and ranking. For retrieval, firstly an inverted index is established on the database, then we can use BM25, keyword, or embedding to recall the entities. While in our e-commerce situation, the title created by the seller sometimes is very cryptic, leading to the recall result "low relevance" or "less results".

In order to overcome these problems, we first filter out the valid entity based on the popularity of the entities to limit the number of it within a valid range of multi-classification, then use rough ranking module and exact ranking module to get matched entities. Finally, we adjust the ranking results of difficult samples.

A large amount of ranking algorithms[2,3,4,5] have been proposed to optimize the ranking performance. There are mainly three types of learning to rank: Pointwise[6,7], Pairwise[8,9], and Listwise[10]. Deep Structured Semantic Model[1] are mostly used for LTR, as well as various improvements based on it, such as CLSM[11],LSTM-DSSM[12]. In recent years, transformer-based model like bert[13] performs well in many NLP tasks, also some variant bert like Bert-wwm-ext[14], Robert[15], etc., It has become a trend to apply model based transformer in industry, such as the reordering model PRM[17]. These works have proved that the introduction of Transformer can achieve good results. We think that the Transform-based structure can achieve a better effect to pointwise LTR. The experiment we conducted bert-based alse confirmed it.

3. METHODOLOGY

3.1 Overview of the Approach

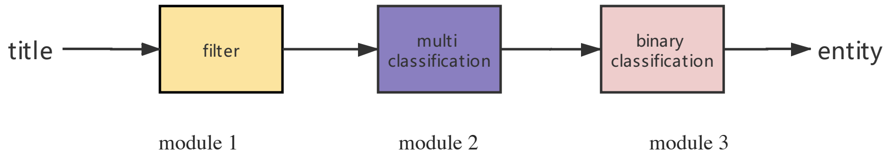

Fig. 1. An overview of our pipline method.

Figure 1 gives an overview of our solution. In module 1, we filter out the valid entity based on the popularity of the products to limit the number of class within a valid range of multi-classification in module 2, then a multiclass classification model based Bi-GRU is trained with the entity as sample's label to get the candidate products by choosing top N highest predicted probability of a sample further. With the candidate entities, we train a binary classification model based bert with all text features of a entity concatenated with the title text to rerank the previous order in module 2. The result with the highest probability is regarded as the matched entity In module 2, for those samples with the highest probability of being predicted to belong to an entity but still below the threshold we set, we consider them as hard samples. For this hard samples, we expand their candidate entities to the all valid entities and predicate again in the module 3 to recorrect the result before.

3.2 Data preprocessing

3.2.1 Data cleaning

First of all, there are many obscure expressions in the title text which seller created to avoid the risk in the platform, such as punctuation marks or html symbols in the title text. We delete this special kind of symbols and convert traditional Chinese to simplified Chinese. In all, we keep only Chinese, English, and Numbers.

3.2.2 Entity information construction

In the product library, each entity contains the entity name, factory, ingredients, efficacy, origin, specifications, etc. like that, and we construct the complete description of a entity by connecting all predicate and object together.

```
{
'type': 'Medical',
'subject_id': 51730,
'subject': '盾叶冠心宁片',
'data': [
        {'predicate': '生产企业', 'object': '江苏黄河药业股份有限公司'},
        {'predicate': '主要成分', 'object': '盾叶薯蓣的根茎提取物。'},
        {'predicate': '症状','object': '活血化瘀，行气止痛、养血安神。用于治疗胸痹、心痛属气滞血瘀症、高脂血症以及冠心病、心绞痛见上述证候者。对胸闷、心悸、头晕、失眠等症有改善作用。'},
        {'predicate': '规格', 'object': '铝塑板装，2×12 片/板/盒。'},
        {'predicate ': '产地', 'object': '中国'}]
}
```

芬太尼口腔片，生产企业:常州四药制药有限公司，主要成分:芬太尼，症状:本品用于治疗需要应用阿片类止痛药物的重度慢性疼痛。规格:100mg/片，产地:美国

3.3 Filter

In general, entity search often retrieve entity from entity library by constrcting rules like bm25, but in our scenario, the title text of many samples does't have obvious semantic association with the entity they refer to, thus leading to the recall result "low relevance" or "less results".

We have observed that in actual data the product entities appearing on the e-commerce platform obey the obvious long-tail distribution, that is to say, a small number of entities constitute most of the samples in the training set, so we measure the popularity of the commodity entity by the frequency of they appearing in the training set, and we select out effective entities of high-frequency from the large entity library.

3.4 Multi-calssification

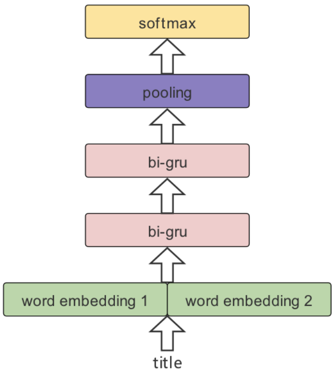

Fig. 2. The architecture of the multi classification model.

In this model, we train a simple multiclass classificationmodel with only using text feature of title to roughly rank the recall results in module1. Fig. 2 illustrates the multiclass classification model based on structure of two-layer BI-GRU. The text of product title is fed to the model as input and the corresponding entity as the category label. The Top N entities with the highest probability are filterd out as candidate results in the next module. The number of categories is limited to the number of valid entities filtered by the module 1, so that under the premise of accuracy, we reduce the number of candidate entities for exact-ranking in module 3 and reduce the time of exact-ranking.

3.5 Binary classification

In the module 3, we rank the candicate entities with regarding the task as a binary classification task of discriminating the correlation between title and subject. The structure of the model is shown in Figure 3.

We exact tune bert as a binary classification model in this module for transformer's outstanding perform in NLP task. For the input, we spliced the title text and content of entity we constructed in 4.2.2 with [SEP] to express the two kinds of information from different sources. All entities and title information are fully utilized in this module. We have made slight adjustments to the structure of bert so that it better fits our tasks. A full-connected layer is spliced to bert with weighted add the outputs of 12-layer of bert as input. For the full-connected layer, we use the multisample dropout structure[16] which helps speed up the training and improves the generalization of the model. The final output of the model is a probability calculated by sigmoid which expresses the probability of the title text describes the entity it concated. Temporarily, we regard the entity with the highest prediction probability as the matched entity.

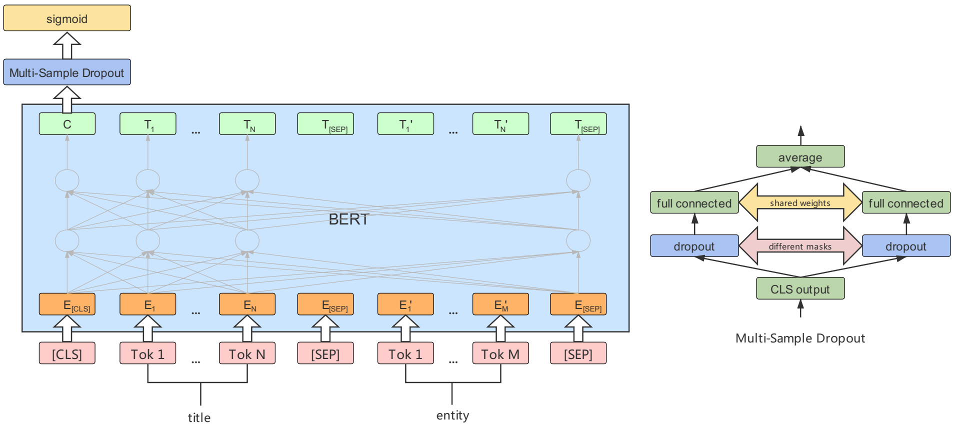

Fig. 3. The architecture of the binary classification classification model.

3.6 Hard sample

For the samples whose highest probability of belonging to a entity is still below a threshold, we believe that the recall is not successful in the module 2 for the unbalanced training set makes the model fit the distribution and ignores the semantic learning of small categories. For these samples, we replace its candidate enetities by all valid entities to predicte it in module 3 again, thus modifying the result o if module 3.

4. Experiment

We conduct experiments on the daset of CCKS 2020: Product Entity Query. The dataset is provided by the Alibaba Inc which includes 83k train samples with text titles and the entity id. Supplementally, 277k complete entities information are also provided. Each Entity information consists of subject id, subject name, and other feature pairs. The task aims to matching entity of the title text refer to. The accuracy of matched eneity is used as the evaluation metric.

4.1 experiment effect

Module	Accuracy on test b
Multiclass classification	0.84528
Binary classification	0.86503
Correct hard sample	0.87947

Table 1 Accuracy of the three modules

Module 2	Accuracy on test b
Text-CNN	0.84018
FastText	0.84250
LSTM	0.83120
Capsule	0.84433
Bi-GRU	0.84520

Table 2 Accurary of different models in module 2

Module 3	Accuracy on test b
alBert-Tiny	0.8364
Bert-wwm-ext	0.8492
Bert-base	0.8565
Roberta-wwm-ext （epoch 5）	0.8582
Roberta-wwm-ext （epoch 7）	0.8650

Table 3 Accurary of different models in module 3

We build our model as described in the chapter 3, the results demonstrate that our pipline is an effective way. Table 1 shows that we take adventage of all entity information in the binary classification model to bring an relative improvement of more than 2.3% compared to the multiclass classification model. Table 2 and Table 3 show the result of module 2 and module 3. For the hard sample, expanding their candicated entities brings a absolute improvement of 1.6% or more compared to the previous.

5. CONCLUSION

In this paper, we present a pipline solution which includes filter, rough ranking, exact ranking, and hard samples supplement to solve the entity match of title-based in large-scaled entity set. We show that our pipline method can achieve SOTA at the accuracy in the the task.

6. REFERENCES

[1]. Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2013: 2333-2338.

[2]. Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning. ACM, 89–96.

[3]. Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. Learning 11, 23-581 (2010), 81.

[4]. Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning. ACM, 129–136.

[5]. Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). ACM, New York, NY, USA, 7–10. https://doi.org/10.1145/2988450.2988454

[6]. David Cossock and Tong Zhang. 2008. Statistical analysis of Bayes optimal subset ranking. IEEE Transactions on Information Theory 54, 11 (2008), 5140–5154.

[7]. Ping Li, Qiang Wu, and Christopher J Burges. 2008. Mcrank: Learning to rank using multiple classification and gradient boosting. In Advances in neural infor- mation processing systems. 897–904.

[8]. Christopher J Burges, Robert Ragno, and Quoc V Le. 2007. Learning to rank with nonsmooth cost functions. In Advances in neural information processing systems. 193–200.

[9]. Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 133–142.

[10]. Thorsten Joachims. 2006. Training linear SVMs in linear time. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 217–226.

[11]. He X, Gao J, Deng L, et al. Convolutional latent semantic models and their applications: U.S. Patent 9,477,654[P]. 2016-10-25.

[12]. Palangi H, Deng L, Shen Y, et al. Semantic modelling with long-short-term memory for information retrieval[J]. arXiv preprint arXiv:1412.6629, 2014.

[13]. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[14]. Cui Y, Che W, Liu T, et al. Pre-training with whole word masking for chinese bert[J]. arXiv preprint arXiv:1906.08101, 2019.

[15]. Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[16]. Inoue H. Multi-sample dropout for accelerated training and better generalization[J]. arXiv preprint arXiv:1905.09788, 2019.

[17]. Pei C, Zhang Y, Zhang Y, et al. Personalized re-ranking for recommendation[C]//Proceedings of the 13th ACM Conference on Recommender Systems. 2019: 3-11.