Improving Semantic Matching via Multi-Task Learning in E-Commerce

Hongchun Zhang
hongchun.zhc@alibaba-inc.com
Alibaba Group
Hangzhou, China

Tianyi Wang
joshua.wty@alibaba-inc.com
Alibaba Group
Hangzhou, China

Xiaonan Meng*
xiaonan.mengxn@alibaba-inc.com
Alibaba Group
Hangzhou, China

Yi Hu
erwin.huy@alibaba-inc.com
Alibaba Group
Hangzhou, China

Hao Wang
longran.wh@alibaba-inc.com
Alibaba Group
Hangzhou, China

ABSTRACT

Semantic matching plays a critical role in an e-commerce search engine, while one of the biggest challenges is the language gap between queries and products. Traditionally, some auxiliary functions, such as the category navigation, are designed to help buyers to clarify their intent. Recently, the advances in deep learning provide new opportunities to bridge the gap, however, these techniques suffer from the data sparseness problem. To address this issue, in addition to the click-through data from buyers, we exploit other types of semantic knowledge from the product category taxonomy and sellers' behavior. We investigate the correlation between query intent classification and semantic textual similarity, and propose a multi-task framework to boost their performance simultaneously. Moreover, we design a Progressively Hierarchical Classification (PHC) network architecture with the taxonomy to solve the category imbalance problem . We conduct extensive offline and online A/B experiments on a real-world e-commerce platform, and the results show that the proposed method in this paper significantly outperforms the baseline and achieves higher commercial value.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**; **Computational advertising**; *Information retrieval*; • **Computing methodologies → Natural language processing**; **Machine learning**.

KEYWORDS

E-Commerce, Multi-Task Learning, Semantic Matching

ACM Reference Format:
Hongchun Zhang, Tianyi Wang, Xiaonan Meng, Yi Hu, and Hao Wang. 2019. Improving Semantic Matching via Multi-Task Learning in E-Commerce. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2019 eCom).* ACM, New York, NY, USA, 7 pages.

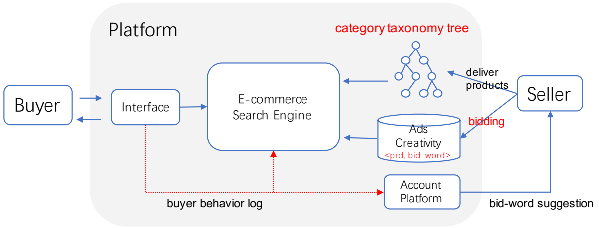


Figure 1: Overview of the e-commerce business ecosystem.

1 INTRODUCTION

Nowadays, buyers increasingly rely on the e-commerce search engine to find their desired products. Similar to the web search, one of the biggest challenges to retrieve relevant products for a query is the language gap problem, when buyers and sellers use different vocabularies to express the same meaning. Traditionally, some auxiliary functions, such as the category navigation, are designed to help buyers to clarify their intent in more details. For example, when a buyer input a query q="`car light`", firstly, the two different products, p1="`auto halogen bulb`" and p2="`led lamp for toy car`" which are both relevant, may be difficult to get recalled by classical text matching; secondly, most buyers who like the first may need another action to filter out the second by choosing an intent category. So, how to predict the query's intent category and address the language discrepancy problem between queries and products are crucial to improve matching efficiency.

Recently, many deep neural networks have been successfully applied to classification [6, 11, 23, 26, 28], and also provide new opportunities to learn better distributed representations of words and sentences to bridge the language gap. However, training a state-of-the-art deep neural network model usually requires a large amount of labeled data which is not always readily available. In a commercial web search engine, it's popular to use click-through data as implicit feedback label [8, 16, 21]. Unfortunately, the data in e-commerce is biased and noisy, because the buyer's click behavior is influenced not only by the recall rate of the online algorithm but also by the product snapshot [24].

Actually, as shown in Figure 1, in addition to the buyer's demand side behavior, there are many other types of semantic knowledge hidden in the platform and the seller's supply side:

[*] Corresponding author.

SIGIR 2019 eCom, July 2019, Paris, France

(1) *product and category*: The platform builds a large-scale taxonomy. Products delivered by sellers are classified into a suitable leaf category. The similarity between products with same category is much higher than that of different categories. However, as shown in Figure 2, the distribution of category data is extremely imbalanced.
(2) *query and category*: When many buyers search the same query and click the same category's products many times, then the category has a very high probability to satisfy the query intent.
(3) *product and bid-word*: The candidate bid-words are usually the history queries with high page view (PV) or conversion rate. The advertisers should pay for the clicks of bid-words to get more exposure. Meanwhile, the cost guarantees their similarity.
(4) *category taxonomy tree*: The path from root to leaf is a process of subdividing layer-by-layer. The tree distance between two category nodes is positively related to their similarity.

Inspired by these observations, in this paper, we propose a multi-task learning framework for semantic matching with multi-type knowledge from e-commerce ecosystem. We firstly generate distributed representation for each input text with TextCNN [28], and then apply two learning tasks: One is a classification task using data (1) and (2), the category plays a bridge role in intent similarity between queries and products. Moreover, we design a Progressively Hierarchical Classification (PHC) network architecture to enrich the similarity of (4). The other task is a pair-wise semantic textual similarity. Specially, we make use of (3) and click-through data as weakly supervised label, and generate comparison training pairs between titles and their positive/negative queries.

Our contribution can be summarized as follows:

- we propose a multi-task learning framework of query intent classification and semantic textual similarity to improve semantic matching efficiency, and make use of multi-type knowledge from the e-commerce ecosystem to address the data sparseness problem;
- we design a PHC network architecture to solve the category imbalance problem, and enrich the similarity between taxonomy tree nodes simultaneously.
- We conduct extensive offline and online experiments on an e-commerce search engine. The results demonstrate the effectiveness of our framework.

2 RELATED WORKS

In recent years, there have been many works to study deep learning for semantic matching. Depending on the stage of signal matching, these methods can be divided into two categories: *Interaction based* and *Representation based.* The former constructs basic low-level matching signals, and then aggregates matching patterns. For instance, ARC-II [7] and MatchPyramid [17] and Match-SRNN [22] are based on word-level similarity matrix, then different network architectures are applied, such as 2-D CNNs [7, 17], RNNs [22]. KNRM [25] and Conv-KNRM [3] make the interaction between every n-gram pair from two pieces of text and employ a kernel pooling layer. The later, such as DSSM [8], CDSSM [21], ARCI [7], CNTN [19], generates the distributed representation for each input text separately, and then applies a classifier to compute the matching score. Although recent works show that interaction-based methods perform better in multiple text matching tasks, but suffer from the expensive online computational complexity.

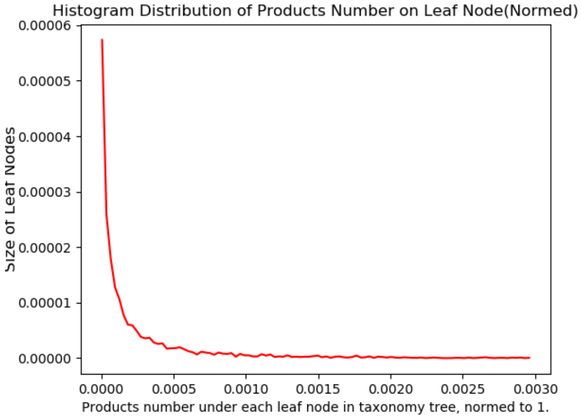


Figure 2: Distribution of products number on category leafs. This is extremely imbalanced.

Moreover, training a deep model needs a large amount of labeled data, which is expensive to obtain. To mitigate this issue, many *unsupervised learning* models seek to exploit the implicit internal structure of the corpus data. For example, various methods for learning distributed word representations, such as word2vec [14], GloVe [18], and sentence representations, such as paragraph vectors [12], skip-thought [10], have been shown very useful for NLP tasks such as sentence classification, sentiment analysis, etc. But it's almost impossible to learn a good representation by unsupervised methods for complex task. Traditionally, *weakly supervised learning* methods are more popular in the industry. DSSM [8], CDSSM [21], LSTM-RNN [16] are trained in a weakly supervised manner with the click-through data. Mostafa et al. [4] used the output of an unsupervised ranking model as a weak supervision signal. Xiao et al. [24] proposed a co-training framework to make use of the unlabeled data. Unfortunately, the click-through data in e-commerce is biased and noisy. Meanwhile, many models address the problem by implicitly performing limited *transfer learning* through the pre-trained embedding of unsupervised methods. Daniel et al. [2] presented a model to learn universal sentence encoder, which specifically targets transfer learning to several NLP tasks. Subramanian et al. [20] explored a *multi-task learning* technique with different training objectives to learn general sentence representation. These works demonstrate that sharing a single sentence representation across related tasks leads to consistent improvements.

In an e-commerce scenario, in addition to the buyers' behavior, the platform and sellers also provide different types of labeled data. Inspired by the weakly unsupervised and multi-task learning methods, in this paper, we investigate query intent classification and semantic textual similarity as two related tasks for semantic matching. Moreover, to solve the category imbalance problem, we design a PHC architecture with the taxonomy category path, which is different from the algorithm-level [13] and data-level methods [1].

3 APPROACHES

In this paper, we illustrate a novel multi-task framework via leveraging product category taxonomy tree to incorporate the correlation between semantic textual similarity and query intent classification using a weakly supervised method for generating training data. We formally define these two tasks at first, and then present our data generation method and modeling paradigm.

3.1 Semantic Similarity

3.1.1 Semantic Similarity (SS). SS is a core problem in many NLP tasks. While in an e-commerce scenario, we also formulate a SS problem as measuring the similarity between a query and all products' titles to retrieve those products which are semantically consistent with the query. We call this task as Query-Title Similarity (QTS).

Traditionally, in the QTS problem, the similarity between each query and its best-matched titles would be calculated. In this work, we implement QTS in a dual way to utilize the knowledge from sellers' side, and calculate the similarity between each title and its best-matched queries instead.

Given a product title t and its candidate queries as set $Q = \{q_i|1 \leq i \leq n\}$, the category labels of t and Q are denoted as c_t and $C_Q = \{c_i|1 \leq i \leq n\}$, s.t. $c_i \in [1, |N_c|]$, $|N_c|$ is the category size. In QTS task, our framework needs to calculate the similarity between t and each q_i in Q, which is defined as $F_s(t, q_i; \theta_s) \to \mathbb{R}^+$, where F_s is a score function and θ_s is the parameter of F_s. Suppose t's best-matched queries could be denoted as $Q_t^+ \subseteq Q$, then others as Q_t^- and $Q = Q_t^+ \bigcup Q_t^-$. Consequently, the objective of QTS task could be designed as minimizing:

$$\log(\frac{\sum_{q_j \in Q_t^+} F_s(t, q_j; \theta_s)}{\sum_{q_k \in Q} F_s(t, q_k; \theta_s)}) \tag{1}$$

But this formulation is impractical because the cost of computational complexity is proportional to sample numbers in $Q^+$and Q, which is often very large in our e-commerce scenario (more than10^8). We would utilize an alternative method to solve it, and refer to 3.1.2 for more details.

3.1.2 Negative Sampling. An alternative method to optimize (1) is using Noise Contrastive Estimation (NCE), which is applied by [5, 15] to language modeling. This strategy is similar to hinge loss which is also trained by telling positive data from noise samples.

To simplify (1), we select only one query q_* from Q^+ and limit the number of negative queries. The NCE has a noise distribution $P_n(\cdot)$ as a free parameter. Inspired by what has been implemented in [14], we randomly chose samples according to the category frequency distribution $U(c)^{3/4}/Z$. It would reduce training time because of its sampling on high frequency categories and the 0.75 power could make the low frequency categories would be sampled more times than 1 power.

Table 1: Structure of Taxonomy Tree of our site

Level1	Level2	Level3	Level4
30+	200+	500+	5000+

Suppose there are K negative queries for each title t, (1) could be rewritten as

$$L_{s_t} = -\log(F_s(t, q_*; \theta_s)) - \sum_{k=1}^{K} \mathbb{E}_{q_k \sim P_n(c)}[\log F_s(t, q_k; \theta_s)], \quad s.t.\ q_* \in Q_t^+, \tag{2}$$

thus, the final loss of QTS task is defined as

$$L_s = \sum_{t \in T} L_{s_t} \tag{3}$$

Now it could be trained via standard gradient descent. We would propose our method to generate Q in Sec. 3.3.1.

3.2 Query Taxonomy

3.2.1 Text Classification. In an e-commerce scenario, query classification (QC) is important to understand buyer's intent to retrieve more related products. In addition to the QTS task, we also introduce a classification problem to infer the c_q, which is equal to maximize the posterior probability $P(c_q|q; \theta_c)$, thus the trained objective function of QC can be written as

$$-\log(P(c_q|q; \theta_c)) \tag{4}$$

Actually, this method could only assign just one category to each query, while products are all arranged via a taxonomy tree (TT) and have several levels of categories. As shown in Table 1, in alibaba.com, products are arranged into four levels, from broad field to specific. For instance, "`balance scooter`" falls under the category 'Sports & Entertainment→Outdoor Sports→Scooters→Self-balancing Electric Scooters'. Consequently, we also need the taxonomy tree to define a query's categories as well.

3.2.2 Progressively Hierarchical Classification. To take into account all different levels of the category path, we design a hierarchical softmax structure named Progressively Hierarchical Classification (PHC) network to leverage the semantic information from root to leaf progressively. Our proposed structure is different from those conventional hierarchical softmax methods, such as in Mikolov *et al.* [14], where hierarchical softmax is used as a speedup technique, and the binary Huffman tree is constructed by samples frequency and could hardly represent the correlation between different leaf nodes. We call the query taxonomy problem as Query Taxonomy Classification Task (QTC).

Suppose the taxonomy of products composed of L layer, each level has its fitting parameters $\theta_c^l, l \in [1, L]$, level l's category is c^l. We implement an unsupervised method to build a large amount of title-category pairs and query-category pairs, denoted as $\tilde{Q} = Q \bigcup T$ respectively and their C_Q and C_T, refer to 3.3.1 for details. As shown in Figure 4, we design a recursive structure which could take all levels of categories before a specific layer l and the original first

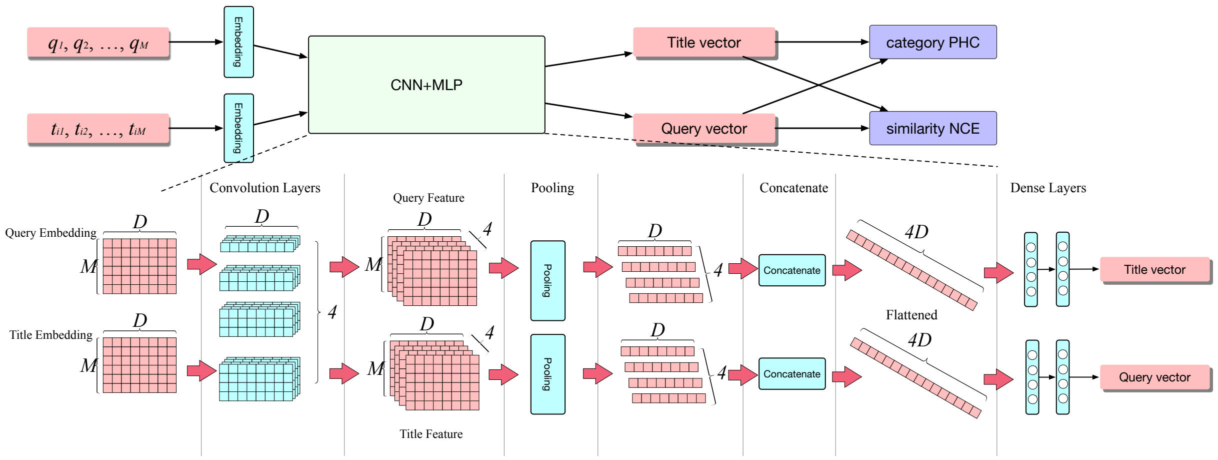


Figure 3: Model Structure: The above half is the whole structure of our framework and another half below is the detailed operations in CNN+MLP block

layer into account and output a category at each level,

$$P(c_q^l|q;\theta_c^l) = \sigma(F_l[\Phi^l(q)]) \tag{5}$$

where

$$\Phi^l(q) = \begin{cases} f_1(q), & l = 1, \\ f_l(f_{l-1}(f_{l-2}(\dots f_1(q)))) + f_1(q), & 2 \le l \le L, \end{cases} \tag{6}$$

where σ is softmax function, $f_l(\cdot)$ refers to no-linear feature mapping at level l and it would output an intermediate representation, and the F_l would project the representation into one-hot category ids. For brevity, we omit parameter in $f_l(\cdot)$. So the final objective function of QTC is

$$L_c = -\sum_{q\in\tilde{Q}}\sum_{l=1}^{L} P(c_q^l|q;\theta_c^l) \tag{7}$$

It might be similar to a recurrent-based decoder for generating TT path because the distribution of category in a layer l always depends on information from previous layer. However, instead of using recurrent neural network to model the category path, we incorporate more parameters $\{\theta_1, ..., \theta_L\}$ which could store more information of the correlation and difference between category nodes. What's more, field knowledge from the taxonomy tree could be preserved more when we add $f_1(q)$ to future levels as a residual than otherwise.

3.3 Multi-Task Learning For Semantic Similarity

In section 3.1 and 3.2, we have presented formulation of the two tasks. In this part, we illustrate our system architecture and show more details on data generation, multi-task strategy and model establishment.

3.3.1 Unlabelled Data Generation. As we mentioned in 3.1, in order to generate enough data for QTS and QTC tasks, we implement an unsupervised method to build a large amount of title-category pairs (T, C_T), query-category pairs (Q, C_Q) and also Q_t^+ for each t.

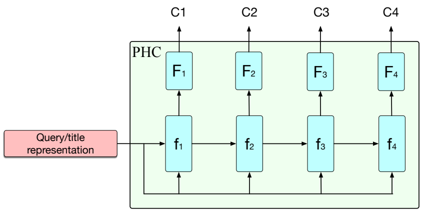


Figure 4: PHC structure

- **title-category pair**. We use the product's category path submitted by sellers as this product title's category path. We defined our product taxonomy as four levels, top level, second level, third level, and leaf category. Without a lack of generality, for those which have less than four levels, we copy their last available category node to fill out the absent ones. In Table 1, we could see the taxonomy size for each level and it's a typical imbalanced distribution, which results from commercial discipline. Therefore, re-sampling from the original data and sample duplication are significant for ensuring the model ability.
- **query-category pair**. Different from title-category pairs, queries have no category path originally. Instead of asking the human to evaluate or annotate millions of queries, we use click-through data as implicit feedback for generating queries' categories [8, 21, 24]. From the click-through data, we filter out high frequency queries firstly and assign the product's category, which has the highest click-through rate, to that query. Also, we use bid-words as queries, and set their binding product's category as training label.
- Q_t^+ **to each title**. Similar to what we do with query-category pairs, firstly we screen out query q and its corresponding products, whose click-through rate is higher than a threshold to form some part of $< q^+, t >$ pairs. We also extract bid-words for each title to form another part of $< q^+, t >$

pairs. Secondly, we randomly take K samples following the distribution $P_n(c)$ as negative queries.

3.3.2 Model Architecture. Inspired by prior works on multi-tasks and co-training [24, 27], we use a multi-task strategy to optimize QTS and QTC tasks simultaneously, which are defined before. The flowchart of whole system is illustrated in Figure 3, we employ a semantic embedding network at the base of whole model, which is similar to TextCNN [9] on the architecture.

Our model would embed the query and title firstly, as shown in Figure 3, and then multi-filter-size convolution blocks would extract specific features at different granularities, from uni-gram to 4-gram, to cover more types of correlation between words and phrases. Mean-pooling is applied after each convolution block and outputs a sentence level representation. The representations of all convolution blocks are concatenated together, forming a comprehensive vector, then two MLP layers are appended behind to project this vector to semantic representation. Because we do not focus on model structure designing in this paper, so for brevity, we define F_s to represent all operations above as

$$F_s(t, q_j; \theta_s) = \text{NormDist}_{cosine}(\Phi^1(t), \Phi^1(q_j)) \tag{8}$$

in which $\text{NormDist}_{cosine}(\cdot)$ is scaled cosine function,

$$\text{NormDist}_{cosine}(v_1, v_2) = 0.5 * \frac{v_1^T v_2}{|v_1||v_2|} + 0.5 \tag{9}$$

where $\Phi(*) \rightarrow \mathbb{R}^d$, $\Phi(\cdot)$ represents the whole non-linear mapping part of embedding model, and our proposed PHC takes $\Phi(t), \Phi(q_j)$ as input and outputs $\{c_t^l\}_{l=1}^L, \{c_{q_j}^l\}_{l=1}^L$. Now the training objective of our multi-task framework could be denoted as

$$L = \lambda_s L_s + \lambda_c L_c \tag{10}$$

where λ_s and λ_c are hyper-parameters. We set λ_s as 1, and λ_c as 0.1 in experiments.

4 EXPERIMENTS

4.1 Dataset and Metric

4.1.1 Dataset. As we introduced in Sec. 3.3.1, our multi-tasks framework needs $(T, C_T), (Q, C_Q)$ and Q_t^+ to train QTS and QTC tasks jointly. Each sample is composed of a triplet of (q^+, Q^-, t), where $Q^- = \{q^-\}$. We build the unlabeled triplets dataset D = $\{(q^+, Q^-, t)\}$ by first sampling search queries and click data from 1-year logs and then generating 10 candidates q^- for every query q^+, t. In total, we get an unlabeled dataset consisting of about $5e8$ (q^+, Q^-, t) triplets. In order to evaluate the semantic representation performance of QTS task, 33,188 <q,t> pairs were annotated into two categories, correlated and uncorrelated, via human effort or user's click-through data as well. Also, queries of these 30,000+ pairs were assigned its category path by human for QTC task too.

4.1.2 Metric. Our multi-task semantic similarity is composed of two tasks: QTS and QTC. In QTS, these pairs which have a similarity score higher than threshold would be assigned 1, the others are 0. So we utilize the classical AUC score to determine the effectiveness. In QTC, we use the accuracy to judge the classifier at each level.

4.2 Implementation Details

In order to prove that improvement of performance on semantic similarity and query classification could be achieved simultaneously and they mutually boost each other, we design a joint-training experiment and also other ablations. We compare our work with following methods:

(i) TextCNN [9] + QTS
(ii) TextCNN* + QTS
(iii) TextCNN* + QTS + QTC w/o PHC
(iv) TextCNN* + QTC with PHC
(v) TextCNN* + QTC w/o PHC
(vi) TextCNN* + QTS + QTC with PHC

TextCNN* represents a classic TextCNN model which is initialized via a word2vec [14] embedding. QTC w/o PHC means that this configuration implements a QTC task but only uses the leaf category in TT and drops the PHC structure. If the configuration has no QTS task, then it is only a taxonomy classification model. We use (ii) as our baseline. In these experiments, we use L=4 to construct a PHC structure with four levels. The TextCNN's embedding size is $80 * V$, where V= 900,000 is the vocabulary size. After that, there have 2 × 4 convolution layers behind, four sets of kernels with lengths from 1 to 4 respectively and we applied max/mean-pooling after each convolution layer at each set. There are also two fully-connected layers behind with size of [128, 128]. $f_l(\cdot) \rightarrow \mathbb{R}^{128}, s.t. 1 \le l \le L$.

4.3 Results Analysis

4.3.1 Performance of Multi-Task Training. We evaluate AUC on the annotated dataset, and test all levels of accuracy too. From Table 2, we can see that TextCNN + QTS has the lowest score on AUC. Since TextCNN* improves a lot on AUC, word2vec embedding initialization is significant. If we take a comparison between results from (ii) (iii) (vi) and (iv) (vi), in which exp. (vi) increases relatively 5.10% and 11.27% at AUC comparing to (iii) and (ii) respectively, at the same time, exp. (iii) and (vi) all perform better than other single-task solutions. At Acc4, (vi) outperform (iv) by 0.87% and (iii) outstrip (v) 4.77%. So it is easy for us to conclude that QTS and QTC are collaboratively optimized and jointly-training them could remarkably enhance each other. As mentioned in the introduction, it could force the category information to flow back into semantic representation and promote its ability to restore more knowledge about categories, which could be intuitively inferred through our observation on data structure in alibaba.com's e-commerce platform. What's more, for all methods, we test their text embedding and illustrate the ROC curve to directly depict their difference of effectiveness. In Figure 5, our proposed method with QTS + QTC + PHC configuration achieve the highest AUC score, which supersedes anyone without multi-task learning.

4.3.2 Gain from PHC structure. From results of (iv) (v) and (iii) (vi), Acc4 gains a 2.51% and 3.40% improvements from (iv) to (v) and (iii) to (vi). These improvements come from the application of the taxonomy tree, in which more levels information of non-leaf layers are restored and also basic field knowledge are strengthened. Additional performance difference between (vi) (iii) and (vi) (ii) prove that with the PHC structure, the QTS could be boosted more. We

Table 2: Metric Scores in QTS and QTC

Model			Similarity Metric AUC	Taxonomy Metric Acc1/Acc2/Acc3/Acc4 (%)
(i)TextCNN	+QTS		0.5557	-
(ii)TextCNN*	+QTS		0.6300	-
(iii)TextCNN*	+ QTS	+ QTC w/o PHC	0.6670	-/-/-/44.39
(iv)TextCNN*		+ QTC with PHC	0.6444	72.44/65.26/58.70/53.23
(v) TextCNN*		+ QTC w/o PHC	0.6193	-/-/-/40.39
(vi)TextCNN*	+ QTS	+ QTC with PHC	**0.7010**	**75.78/65.29/59.26/54.10**

Table 3: Online Common Search Evaluation

SE performance	PV-CTR(%)	FBR(%)	NLS(%)
100% traffic	+1.5	+4.2	-68.7

* SE stands for Search Engine

Table 4: Online Ads Search Evaluation

Ads SE performance	ADS-COV(%)	ADS-CTR(%)	RPM(%)
30% traffic	+1.4	+4.6	+6.7
100% traffic	+4.8	+6.6	+13.4

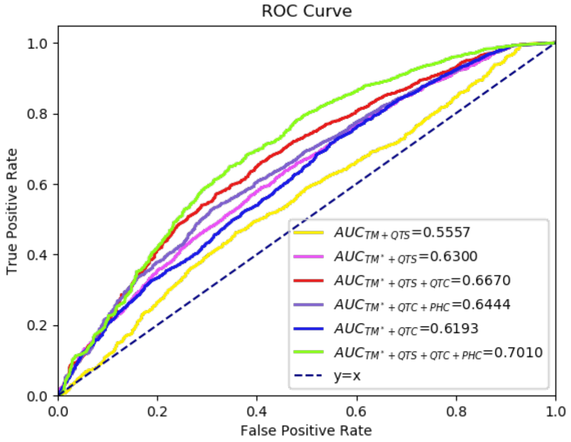


Figure 5: ROC Curve of different Methods

guess all these improvements contributed by adding into relationship between query words and all levels of category-information rather than only the leaf nodes. From Figure 5, we can also see a 4.05% gain from TM* + QTC to TM* + QTC + PHC and a 5.09% from TM* + QTC + QTS to TM* + QTC + QTS + PHC on AUC.

4.4 Analysis on Online Evaluation

4.4.1 Common Search Experiment. For our motivation on semantic matching, we wish to understand user's intent more precisely, retrieve more relevant products and indirectly enhance the Click-Through Rate (CTR) and Feedback Rate (FBR) per page view (PV). In e-commerce scenario, queries without enough retrieval results are called "null and low search" queries (NLS). It would harm the user experience if NLS frequently occur and further undermine the benefit of platform and sellers. So we conduct an online experiments in real-world e-commerce scenario, the www.alibaba.com. We conduct an A/B test based on our multi-task approach inside our online search engine and calculate the NLS rate on tail queries to evaluate the online performance of our methods.

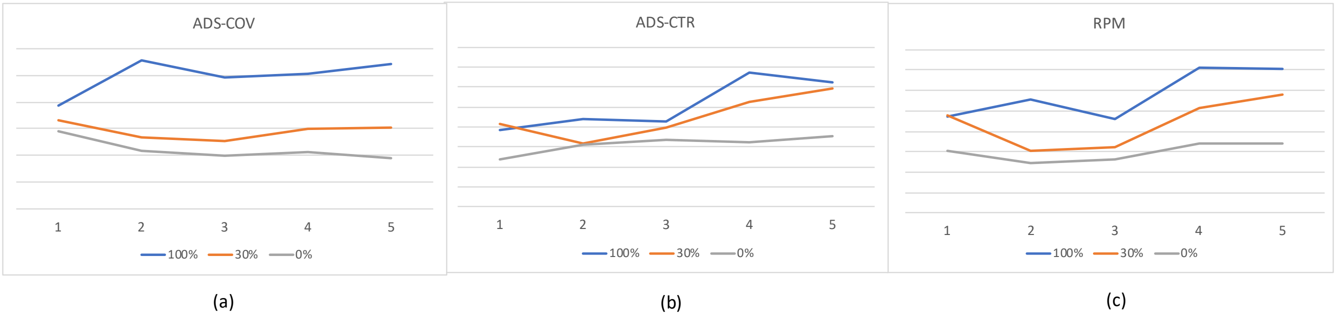
In Table 3, the NLS rate drops 68.7% which is significant while the PV-CTR still increases 1.5%. So we can make the conclusion that our multi-task learning could recall more products and the improved PV-CTR proves that the increments of products are relevant rather than uncorrelated ones which would lead to lower PV-CTR on the contrary. Also the FBR get a gain of 4.2%, which means precision of matching buyer's intent is also increased.

4.4.2 E-commerce Advertisement Experiment. This paradigm used in common search could also be expanded into e-commerce computational advertising scenario (ADS), where the advertisers want their products in advertising campaign to be exposed to buyers with implicit interest for getting orders or feedbacks. In order to prove that our strategy could also enhance the performance in ADS, we implement another online A/B testing experiment on our online ads engine. We use the ads-coverage rate of pv (ADS-COV), Exposure ADS CTR (ADS-CTR) and Revenue per mille (PRM) of platform to evaluate the matching ability. The common metric, ADS-CTR and ADS-COV could be defined as below.

$$ADS - CTR \quad = \quad \frac{N_{\text{ADS-PV}}}{N_{\text{Exposed ADS}}}, \tag{11}$$

$$ADS - COV \quad = \quad \frac{N_{\text{ADS-PV}}}{N_{\text{PV}}}, \tag{12}$$

where the N_* means number of $*$. In Figure 6, we can see that after the Ads SE employed our proposed method, the metric on ADS-CTR, ADS-COV and RPM are all improved significantly. Also, results of continuous five-days online experiment could be found in Table 4. From this result, there is a 13.4% gain on RPM in 100% in search traffic configuration. Also the ADS-CTR is also improved 6.6%, which means that those additional 4.8% ads exposed, which resulting from our strategy are also relatively correlated to buyer's intents.



(a)

(b)

(c)

Figure 6: Five Days Experiment on Ads SE: (a) ADS-COV (b) ADS-CTR (c) RPM

5 CONCLUSION

In this paper, we propose a multi-task method to jointly train query intent classification and semantic textual similarity simultaneously and a novel recursive way to add taxonomy tree into this framework. Experiments show that our proposed strategy could achieve higher accuracy and AUC on classification and similarity problem respectively, which both justify our assumptions that there are positive interaction between these two tasks and using taxonomy tree also can improve semantic representation for queries.

Future work would be carried on two directions: first, we will add more information about products besides titles to improve matching precision to user's query. Secondly, there are many other advanced framework on textual representation, and we would incorporate them into our tasks to obtain more improvement on business.

REFERENCES

[1] Lida Abdi and Sattar Hashemi. 2015. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering* 28, 1 (2015), 238–251.

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[3] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*. ACM, 126–134.

[4] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 65–74.

[5] Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research* 13, Feb (2012), 307–361.

[6] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[7] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*. 2042–2050.

[8] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. ACM, 2333–2338.

[9] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[10] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing systems*. 3294–3302.

[11] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on Artificial Intelligence*.

[12] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neurl Information Processing Systems*. 3111–3119.

[15] Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426* (2012).

[16] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward. 2015. Deep sentence embedding using the long short term memory network: analysis and application to information retrieval. *IEEE/ACM Transactions on Audio Speech & Language Processing* 24, 4 (2015), 694–707.

[17] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[19] Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[20] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf* (2018).

[21] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Information & Knowledge Management*. ACM, 101–110.

[22] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378* (2016).

[23] Jing Wang and Min-Ling Zhang. 2018. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2427–2436.

[24] Rong Xiao, Jianhui Ji, Baoliang Cui, Haihong Tang, Wenwu Ou, Yanghua Xiao, Jiwei Tan, and Xuan Ju. 2019. Weakly Supervised Co-Training of Query Rewriting andSemantic Matching for e-Commerce. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 402–410.

[25] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*. ACM, 55–64.

[26] Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017. A generalized recurrent neural architecture for text classification with multi-task learning. *arXiv preprint arXiv:1707.02892* (2017).

[27] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. 2018. Visual search at alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 993–1001.

[28] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).