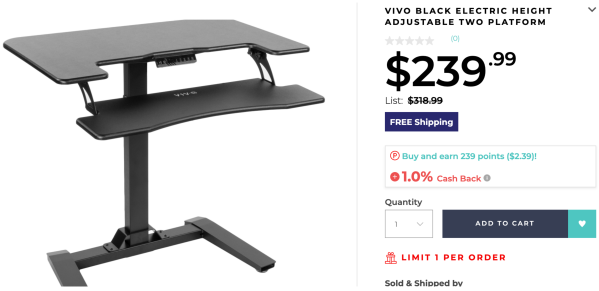
Multi-task Learning for Product Information with Fine-tuned BERT

Chen Zhao　Yuki Nakayama　Koji Murakami
Rakuten Institute of Technology, Rakuten Inc.
{yuki.b.nakayama, chen.a.zhao, koji.murakami}@rakuten.com

1 Introduction

This paper introduces a new technique of deeply analyzing product information in natural text. The problem of tagging product information regarding multiple attributes has been long studied as a sequence tagging task. The problem draws special attention in E-Commerce applications because tagging attribute information like product brand names is particularly useful when product profile information such as titles and descriptions come in large quantities. Once successfully automated, this task allows various product attributes to be recognized from free-form text saving significant labor of human annotation. The difficulty of this problem primarily comes from stacking of attribute complexity and irregular text format. In early days, simple and rule-based heuristics were used such as regular expression (regex) matching for identifying product brands from titles. But regex matching suffers from low recall as unseen brands never get invited to the pre-defined rules. While naive probabilistic approaches like linear chain conditional random field (CRF) achieves better generalization, they lead to unacceptably low precision. In addition, for official E-Commerce Web sites, there are downstream tasks including item classification, purchase behavior analysis and end-user recommendation, all relying upon accurate product information.



VIVO BLACK ELECTRIC HEIGHT ADJUSTABLE TWO PLATFORM

(0)

$239.99

List: ~~$318.99~~

FREE Shipping

Buy and earn 239 points ($2.39)!

1.0% Cash Back

Quantity

1

Figure 1: Screenshot of an Office Desk

With the help of modern named entity recognition (NER) [1] [2] [7], a substantial amount of related works [9] [5] have been done on improving quality of tagging product information with machine learning. Most noticeably, neural architectures combined with CRF as an output layer are proved by various studies [8] [4] [6] to achieve better overall performance than all the conventional heuristics and methods in early days. Still these approaches generalize poorly in *open world assumption* which imposes no limitation on vocabulary of specific product aspect, such as a brand name. *OpenTag* [10] achieves considerable breakthrough concerning open world assumption such that emerging brand names (i.e., absent in training data) can be correctly found by a model. Nevertheless, all of the aforementioned works are based on similar problem formulation as NER, which becomes problematic as product diversity increases. With a random marketplace item as an example shown in Figure 1, it is commonly required that all its product attributes are predicted from its title and description. The attribute values expected from the example item are listed alongside its unorganized raw text displayed in Figure 2. For typical NER problems, individual sequence labeling models have to be trained for tagging all the 6 highlighted spans. This leads to an explosive number of models when product attributes come in a few thousands and may constantly grow. In this paper, we propose a BERT-based technique that enables multi-task learning on multiple product attributes. Instead of the standard NER problem setting commonly seen in existing works we reformulate multi-task learning as a question answering problem, in which task names (e.g., brand name) are jointly aligned with product information text at the training phase. A fine-tuned model is capable of handling multiple tasks depending on the input task name, or the “question” it receives.

The rest of this paper makes formal definition of the problem and introduces the proposed architecture. The experiments first cover comparison between the proposed approach and NER solutions including BERT-CRF and BiLSTM models. Then experiments on a few large data sets are discussed. Our evaluation metrics indicate that the proposed multi-task learning model suffers no performance loss and works reasonably well on large data sets.

2 Problem Definition

Given a query $\boldsymbol{q}$ representing a task name as the token list $(q_1, ..., q_m)$ and context paragraph $\boldsymbol{c}$ as $(c_1, ..., c_n)$, we want to find a subset of consecutive tokens as the answer, $\boldsymbol{a}$ from $\boldsymbol{c}$. More concretely, this requires a pair of indices to get predicted, the answer starting at s and ending at e, i.e., $\boldsymbol{a} = \boldsymbol{c}_{s:e}$ inclusively, $1 \leq s, e \leq n$ and $s \leq e$. An *example* in this paper is defined as a triplet of $(\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a})$ where $\boldsymbol{a}$ is a valid answer to the query. Two examples are considered different examples unless all the three elements are identical. An *item* is defined as a distinct context paragraph and briefly notated as $\boldsymbol{c}$. Two items are not considered the same if the context paragraphs do not match, even if they belong to identical product. Most commonly, an item contains multiple task names $\boldsymbol{q}$ and ground truth labels $\boldsymbol{a}$ as is illustrated by Figure 2. This also implies that the number of total examples will be much higher than total items in the real data set.

3 Architecture: Fine-tuning and Inference

We explore the Bidirectional Encoder Representations from Transformers (BERT) [3], a recent state-of-the-art language representation model that is pre-trained in an unsupervised pattern upon massive amounts of corpora. A pre-trained BERT model can be used as a transfer learning checkpoint, which can further be fine-tuned for more specific downstream tasks including classification, NER and machine reading comprehension (MRC). For the purpose of multi-task learning for product attributes, we formulate our problem as MRC[1]. In MRC setting, the query $\boldsymbol{q}$ is aligned with some item $\boldsymbol{c}$ as initial input, as is shown in Figure 3.

Consider an example $\boldsymbol{t} = (\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{a}) = (\boldsymbol{q}, \boldsymbol{c}, \boldsymbol{c}_{s:e})$. The ground truth of the given example is $\boldsymbol{y} = (s, e)$. Let the initial feature of the example be $\boldsymbol{x}$, the model input takes the form $\boldsymbol{x} = ([\text{CLS}], q_m, [\text{SEP}], ..., q_1, [\text{SEP}], c_1, ..., c_n), \boldsymbol{x} \in \mathbb{R}^{L \times H'}$, where [CLS] is a dummy token indicating the start of input sequence and D is the input embedding size. Notice that the query tokens are in inverse order and separated by a separation token. This allows multi-token queries to be better recognized to improve overall performance verified by experiments in the next section. BERT encodes input features $\boldsymbol{x}$ into an output sequence $\boldsymbol{h}(\boldsymbol{x}), \boldsymbol{h} \in \mathbb{R}^{L \times H}$ where L is the same as input sequence length and H is the inner dimension of BERT output. For the downstream task of span prediction, a fully connected output layer is appended, so that the output tensor $\boldsymbol{l} = \boldsymbol{h}(\boldsymbol{x})W + b, W \in \mathbb{R}^{H \times 2}, b \in \mathbb{R}, \boldsymbol{l} \in \mathbb{R}^{L \times 2}$. The two columns in $\boldsymbol{l}$ after softmax are considered starting and ending probabilities of the answer: $\boldsymbol{l}_1 = softmax(\boldsymbol{l}[:, 0])$ and $\boldsymbol{l}_2 = softmax(\boldsymbol{l}[:, 1])$. Then for feature $\boldsymbol{x}$ we compute the reduced sum of two cross-entropy losses $\mathcal{L}(\boldsymbol{x}) = -\frac{1}{2}[I_s log \boldsymbol{l}_1 + I_e log \boldsymbol{l}_2], I_s, I_e \in \mathcal{R}^L$, where I_s and I_e are one-hot vectors at positions s and e.

Next we define an augmented feature $\tilde{\boldsymbol{x}} = ([\text{CLS}], [\text{MASK}], [\text{SEP}], ..., [\text{MASK}], [\text{SEP}], c_1, ..., c_n)$ by replacing all the query tokens with [MASK] in $\boldsymbol{x}$. Feature $\tilde{\boldsymbol{x}}$ goes through the same computation layers and similarly we have two softmax vectors $\tilde{\boldsymbol{l}}_1$ and $\tilde{\boldsymbol{l}}_2$. For the augmented feature, we compute the loss $\tilde{\mathcal{L}}$ only based on the first elements. So let $\tilde{l}_1 = \tilde{\boldsymbol{l}}_1[0]$ and $\tilde{l}_2 = \tilde{\boldsymbol{l}}_2[0]$. We define $\tilde{\mathcal{L}}(\tilde{\boldsymbol{x}}) = -\frac{1}{2}[log\tilde{l}_1 + log\tilde{l}_2]$.

Finally we maintain a count set for distinct queries. Let $\boldsymbol{q}(\boldsymbol{t})$ be the query of example $\boldsymbol{t}$ and Q be the set of all queries. The query count $C(\boldsymbol{q})$ is the number of examples with $\boldsymbol{q}$ in training data.

$$C(\boldsymbol{q}) = \left\| \{ \boldsymbol{t} \mid \boldsymbol{q}(\boldsymbol{t}) = \boldsymbol{q}, \boldsymbol{q} \in Q \} \right\|$$

Now we are ready to define the final training loss subject to minimization for the example $\boldsymbol{t}$.

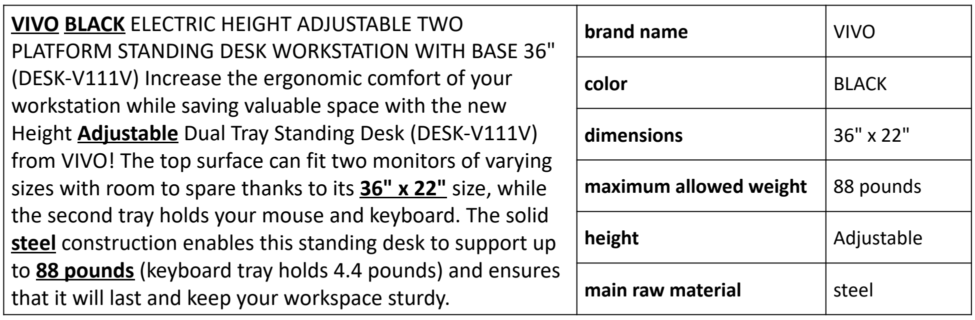
$$\mathcal{L}_{train}(\boldsymbol{t}) = \mathcal{L}(\boldsymbol{x}) + \sqrt{\frac{C_{min}}{C(\boldsymbol{q})}} \tilde{\mathcal{L}}(\tilde{\boldsymbol{x}}) \quad (1)$$

where C_{min} is the lowest count among all the queries: $C_{min} = min\{C(\boldsymbol{q}) | q \in Q\}$. For given training data, all $C(\boldsymbol{q})$ are easily precomputed. The square root coefficient is designed to impose penalty against examples with rare queries by increasing the loss on its augmented features $\tilde{\boldsymbol{x}}$. For examples with frequent queries, the latter term in (1) tends to be negligible as the denominator can be a few thousand times larger than the numerator. Such design proves beneficial for model performance as is discussed in the next section. For model inference, we simply input a feature $\boldsymbol{x}$ and after the softmax layer receive $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$. Hopefully the model predicts a correct answer as $\hat{\boldsymbol{y}} = (\hat{s}, \hat{e})$, where $\hat{s} = \underset{i}{\text{argmax}}\, \boldsymbol{l}_1$ and $\hat{e} = \underset{i}{\text{argmax}}\, \boldsymbol{l}_2$. The final answer in textual format is the span $\boldsymbol{c}[\hat{s} : \hat{e}]$.

4 Experiments

The experiments in this paper are designed to showcase two facts. First, we prove through a set of comparative experiments that the proposed technique is fully compatible with single-task learning

[1] In this paper, the only type of model involved is BERT-Base, English, uncased version with 110M parameters.



VIVO **BLACK** ELECTRIC HEIGHT ADJUSTABLE TWO PLATFORM STANDING DESK WORKSTATION WITH BASE 36" (DESK-V111V) Increase the ergonomic comfort of your workstation while saving valuable space with the new Height **Adjustable** Dual Tray Standing Desk (DESK-V111V) from VIVO! The top surface can fit two monitors of varying sizes with room to spare thanks to its **36" x 22"** size, while the second tray holds your mouse and keyboard. The solid **steel** construction enables this standing desk to support up to **88 pounds** (keyboard tray holds 4.4 pounds) and ensures that it will last and keep your workspace sturdy.

brand name	VIVO
color	BLACK
dimensions	36" x 22"
maximum allowed weight	88 pounds
height	Adjustable
main raw material	steel

Figure 2: Aligned Product Information

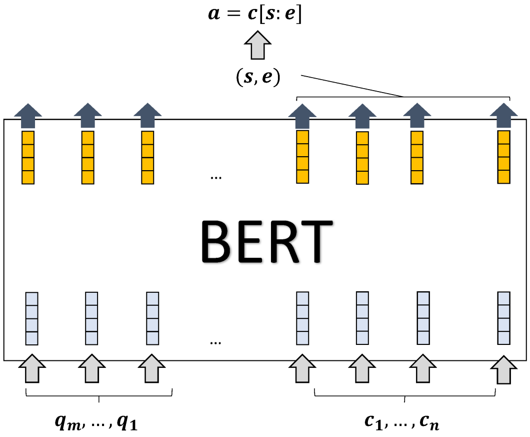


Figure 3: Model Overview

and does not lose model accuracy compared to popular sequence labeling techniques including BERT-CRF and BiLSTM-CRF. Second, experiments on 3 large data sets consisting of examples covering a range of different product attributes. Examples are taken from real world item titles/descriptions and product attribute information from Rakuten's English market, *rakuten.com*.

Five independent tasks are picked out for comparative experiments. Each task corresponds to a separate data set in which all examples share a common query, as is listed in Table 1. We picked up data of varying sizes to better challenge the models on different scales. For each data set, approximately 10% random examples are held out as test data. Then BERT-CRF, BiLSTM-CRF and the proposed multi-task BERT are trained and evaluated on the same train/test data. For all BERT based models in comparative experiments, the maximum sequence length is set at 128 with learning rate of $3e{-}5$. Training batch size is 12 and all training undergoes 2 epochs. Evaluation outcomes in Table 1 show that the proposed multi-task learning method outperforms both BERT-CRF and BiLSTM-CRF in terms of both exact match and F1 measure. Here exact match score refers to percentage of perfectly matched predictions among all test examples.

As for experiments on large data sets, we focus on 3 product categories. Similarly we randomly held out 10% examples from entire data as test sets. For each data set parallel experiments are conducted on vanilla BERT[2], BERT with query token inversion without augmented loss and the final approach as is proposed in Section 3. Table 4 illustrates the effectiveness of the proposed methods albeit minor improvement, compared to unmodified vanilla BERT.

5 Conclusion

In this paper, we proposed a novel strategy of fine-tuning BERT to optimize model accuracy for multi-task learning of product attributes. Experiments show that our solution can replace single-task NER solutions without adversely affecting overall performance. Further experiments illustrate its success on large data sets containing multiple tasks.

References

[1] Dan Bikel, Richard Schwartz, and Ralph Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34, 02 1999.

[2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[2] We comply with configuration used by Devlin et al. for SQuAD MRC tasks

Table 1: Evaluation - Comparative Experiments

Task Name	# Examples	Metrics	BERT-CRF	BiLSTM-CRF	Multi-task BERT
color	454,268	exact f1	91.11 95.75	91.02 95.45	**97.89** **98.29**
brand name	10,847	exact f1	89.21 92.26	80.63 85.55	**92.90** **94.71**
operating system platform	157,346	exact f1	99.63 99.63	98.22 98.25	**99.79** **99.79**
operating system language	29,819	exact f1	83.76 84.01	95.40 95.47	**98.26** **98.26**
graphics controller model	42,310	exact f1	90.32 91.25	87.86 89.90	**95.21** **95.80**

Table 2: Evaluation - Large Data Experiments

Category	# Examples # Queries	Metrics	Vanilla BERT	BERT + query inversion	BERT + query inversion + [MASK]ed features
consumer electronics	700,547	exact	86.97	**87.09**	87.01
	256	f1	87.91	87.94	**87.99**
home and living	525,872	exact	88.23	88.64	**88.66**
	275	f1	90.80	90.93	**90.98**
connector cables	748,172	exact	98.20	98.17	**98.20**
	26	f1	98.52	98.49	**98.55**

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[5] Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. Recognizing salient entities in shopping queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 107–111, Berlin, Germany, August 2016. Association for Computational Linguistics.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[7] Xiao Ling and Daniel S. Weld. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI ’ 12, page 94–100. AAAI Press, 2012.

[8] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

[9] Duangmanee Putthividhya and Junling Hu. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[10] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD ’ 18, page 1049–1058, New York, NY, USA, 2018. Association for Computing Machinery.