# Extreme Multi-Label Classification with Label Masking for Product Attribute Value Extraction

**Wei-Te Chen**　**Yandi Xia**　**Keiji Shinzato**
Rakuten Institute of Technology,
Rakuten Group Inc.
`{weite.chen, yandi.xia, keiji.shinzato}@rakuten.com`

## Abstract

Although most studies have treated attribute value extraction (AVE) as named entity recognition, these approaches are not practical in real-world e-commerce platforms because they perform poorly, and require canonicalization of extracted values. Furthermore, since values needed for actual services is static in many attributes, extraction of new values is not always necessary. Given the above, we formalize AVE as extreme multi-label classification (XMC). A major problem in solving AVE as XMC is that the distribution between positive and negative labels for products is heavily imbalanced. To mitigate the negative impact derived from such biased distribution, we propose label masking, a simple and effective method to reduce the number of negative labels in training. We exploit attribute taxonomy designed for e-commerce platforms to determine which labels are negative for products. Experimental results using a dataset collected from a Japanese e-commerce platform demonstrate that the label masking improves micro and macro $F_1$ scores by 3.38 and 23.20 points, respectively.

## 1 Introduction

Since organized product data plays a crucial role in serving better product search and recommendation to customers, attribute value extraction (AVE) has become a critical task in the e-commerce industry. Although many studies have treated AVE as named entity recognition (NER) task (§ 2.1), NER-based approaches are not practical in real-world e-commerce platforms. First, NER-based methods perform poorly because the number of attributes (classes) in e-commerce domains is extremely large (Xu et al., 2019). Second, it is necessary to take a further step to normalize extracted values (*e.g., coral to pink*). To reflect extracted values in actual services, e-commerce platform providers need to convert the values into canonical form by referring their own attribute taxonomy that covers attributes and values for the services. Third, extraction of new values is not necessary in many attributes (*e.g., country of origin*). Since it is rare for new values of attributes other than brands to be introduced to the world, it is sufficient to extract the values defined in the attribute taxonomy.

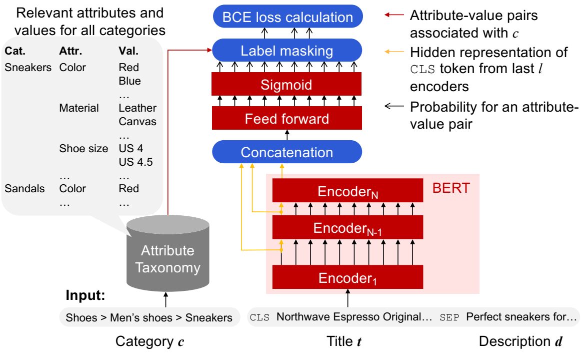
Relevant attributes and values for all categories
Cat.
Attr.
Val.
Sneakers
Color
Red
Blue
...
Material
Leather
Canvas
...
Shoe size
US 4
US 4.5
...
...
Sandals
Color
Red
...
...
Attribute Taxonomy
BCE loss calculation
Label masking
Sigmoid
Feed forward
Concatenation
Encoder_N
Encoder_N-1
Encoder_1
BERT
Attribute-value pairs associated with c
Hidden representation of CLS token from last l encoders
Probability for an attribute-value pair
Input:
Shoes > Men's shoes > Sneakers
CLS Northwave Espresso Original... SEP Perfect sneakers for...
Category c
Title t
Description d

Figure 1: Extreme multi-label classification model with label masking for attribute value extraction.

Given the above reasons, we formalize AVE as extreme multi-label classification (XMC), and design a model that directly predicts possible canonical attribute-value pairs except for brands[1] from given product data. The main problem in solving AVE as XMC is that the number of relevant attribute-value pairs to products is far fewer than that of irrelevant pairs; the majority of attribute-value pairs are regarded as irrelevant (*e.g.,* ⟨*Memory size, 512GB*⟩ *for sneakers*). To tackle this problem, we propose label masking that mitigates the negative effects of a large amount of irrelevant pairs in training (Figure 1, § 4.2). We detect the irrelevant pairs by referring an attribute taxonomy (§ 3) associated with a real-world dataset we use to train and evaluate models. Through experiments using the dataset, we confirm that our label masking method improves micro and macro $F_1$ scores by 3.38 and 23.20 points, respectively.

Our contributions can be summarized as follows:

[1] NER-based methods are necessary to extract new values.

*Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 134 - 140
May 26, 2022 ©2022 Association for Computational Linguistics

- We formalize AVE as an XMC problem.
- We proposed label masking, a simple and effective method to alleviate the negative impact from irrelevant attribute-value pairs in training (§ 4.2).
- We showed the effectiveness of the label masking using a real-world dataset. It especially performed well on attribute-value pairs at the long tail (§ 5.4).

## 2 Related Work

### 2.1 Attribute Value Extraction

There are many attempts based on NER techniques to extract attribute values from product descriptions (Probst et al., 2007; Wong et al., 2008; Putthividhya and Hu, 2011; Bing et al., 2012; Shinzato and Sekine, 2013; More, 2016; Zheng et al., 2018; Rezk et al., 2019; Karamanolakis et al., 2020; Zhang et al., 2020). As Xu et al. (2019) reported, NER-based models perform poorly on a real-world dataset including ten thousand attributes or more.

To deal with a large number of attributes, there is research that introduces question-answering (QA) models for the AVE task (Xu et al., 2019; Wang et al., 2020; Shinzato et al., 2022). These QA-based approaches take an attribute as *query* and a product title as *context*, and extract attribute values from the context as *answer* for the query. Since those models take attributes as input, it is necessary to run the extraction repeatedly on the same product titles with different attributes. Hence, the QA-based approaches are more time-consuming than XMC-based approaches that can predict values for multiple attributes at a time.

### 2.2 Extreme Multi-Label Classification

To reduce the large output space, previous XMC studies perform label clustering as a separate stage from training classifiers (Wydmuch et al., 2018; You et al., 2019; Chang et al., 2020; Zhang et al., 2021; Jiang et al., 2021; Mittal et al., 2021a,b). For example, XR-Transformer (Zhang et al., 2021) first vectorizes each label with combination of TF-IDF and embeddings of text associated with the label. Then, it applies balanced k-means (Malinen and Fränti, 2014) to these label vectors to generate a hierarchical label cluster tree by recursively partitioning label sets. Instead of k-means, Mittal et al. (2021a) and Mittal et al. (2021b) partition labels into equal sized clusters, and then train a binary classifier per cluster that predicts whether a given text is relevant to labels in the cluster.

| Category | Shoes > Men's shoes > Sneakers | | |
|---|---|---|---|
| Attributes | Color | Material | Shoe size |
| Values | • Red<br>• Blue<br>• Green | • Leather<br>• Canvas<br>• Gore-Tex | • US 4<br>• US 4.5<br>• US 5 |

Table 1: Example of attribute taxonomy.

On the other hand, in real-world e-commerce platforms, an attribute taxonomy is available. This can be regarded as label clusters manually tailored by the e-commerce platform providers. Therefore, we simply leverage the existing attribute taxonomy to reduce the size of labels in training through label masking.

## 3 Attribute Taxonomy

We assume that for each category, attribute taxonomy defines all possible attribute-value pairs that products in the category can take. General attribute-value pairs (*e.g.*, ⟨*Color, Red*⟩) are defined for multiple categories. Table 1 shows an example of attributes and values defined for the category of sneakers. By referring to the attribute taxonomy, it is possible to determine which attributes and values are relevant or irrelevant to which category of products. For example, from the table, we can see that 512GB of memory size is irrelevant to sneakers.

## 4 Proposed Method

This section proposes our model based on XMC with label masking for the AVE task. Given a product data $\boldsymbol{x} = \langle c, \boldsymbol{t}, \boldsymbol{d} \rangle$, where $c$ denotes a category, $\boldsymbol{t}$ denotes a title consisting of $n$ tokens ($\{t_1,t_2,\ldots,t_n\}$) and $\boldsymbol{d}$ denotes a description consisting of $m$ tokens ($\{d_1,d_2,\ldots,d_m\}$), respectively, the model returns a set of attribute-value pairs that should be linked with the product data $\boldsymbol{x}$.

Figure 1 depicts the model architecture. As a backbone of the architecture, we employ a pre-trained BERT-base model (Devlin et al., 2019), and put a feed forward layer on the top of BERT. As an input to BERT, we construct a string $[\text{CLS}; \boldsymbol{t}; \text{SEP}; \boldsymbol{d}]$ by concatenating $\boldsymbol{t}$, $\boldsymbol{d}$, CLS and SEP; CLS and SEP are special tokens to represent a classifier token and a separator, respectively. Similar with Jiang et al. (2021), we concatenate the last $l$ hidden representations of the CLS token, and then feed the concatenated vector into a feed forward

| Category | Title | Description | Attribute-value pairs |
|---|---|---|---|
| 靴 > メンズ靴 > スニーカー | ノースウエーブ【northwave】ESPRESSO ORIGINAL RED 男性用 メンズ / 女性用 レディース / スニーカー | 製品説明 落ち着いたレッドが象徴的な、足元のアクセントとして最適な1足。軽量ラバーでソールも軽量化された人気カラーのモデル。 | ⟨ 靴サイズ(cm), 25.0 ⟩,<br>⟨ 靴サイズ(cm), 26.0 ⟩,<br>⟨ 靴サイズ(cm), 27.0 ⟩,<br>⟨ カラー, レッド ⟩ |
| Shoes > Men's shoes > Sneakers | Northwave [northwave] Espresso Original Red Men's / Women's / Sneakers | Product description. These sneakers are the perfect accent for your feet and come in a soft red color. The sole is made of lightweight rubber to reduce weight. It is a popular color. | ⟨ Shoe size (cm), 25.0 ⟩,<br>⟨ Shoe size (cm), 26.0 ⟩,<br>⟨ Shoe size (cm), 27.0 ⟩,<br>⟨ Color, Red ⟩ |

Figure 2: Example of product data. The top shows the original data and the bottom shows its translation.

layer as the representation of the input.

The size of the outputs from the feed forward layer is equal to the total number of labels (attribute-value pairs). The outputs are converted into probability through a sigmoid layer, and then pass to the label masking. To mask labels irrelevant to the given product data $x$, we refer an attribute taxonomy built for an e-commerce platform. We compute binary cross entropy (BCE) loss over only relevant labels.

In testing, we choose ones whose probability returned from the model exceeds 0.5 among labels relevant to the product data $x$.

### 4.1 Preliminary: XMC

XMC is a special case of the multi-label classification problem. What makes XMC unique is its size of a target label set. The label size is 4K to 501K in common XMC datasets (Chang et al., 2020).

Formally, XMC can be defined as follows: Giving a training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ where $x^{(i)}$ is the instance, and $y^{(i)} \in \{0,1\}^L$ is the label of $x^{(i)}$ represented by $L$ dimensional multi-hot vectors. $L$ is the size of the label set. $y_j^{(i)} = 1$ indicates that the $j$-th label is a positive example for $x_i$. The regular XMC is aimed to learn the function $\sigma_\theta(x) \in \{(0,1) \subset \mathbb{R}\}^L$ which predicts scores in range of [0.0, 1.0] to all labels by giving $x$. $\sigma$ tends to be closed to 1.0 to $j$-th label when $y_j = 1$. The ordinary loss function in XMC is BCE:

$$\mathrm{BCE} = -\sum_{j=1}^{L} (y_j \log \sigma_\theta^j(x) + (1-y_j)\log(1-\sigma_\theta^j(x)))$$

BCE loss sums over the log loss among all labels.

### 4.2 Label Masking

In the AVE task, the number of "hot" labels is extremely small compared to the number of labels defined for the task ($L$). This means that distribution between positive and negative labels is heavily imbalanced. Such distribution has the negative impact on training classification models because the BCE sums far more loss values from the negative labels.

To alleviate the impact derived from the negative labels, we exploit attribute taxonomy. Since the majority of the negative labels are irrelevant labels to given product data $x$, we introduce a function $\mathbb{M}$ that returns only relevant labels to $x$. BCE loss can be rewritten as follows:

$$\mathrm{BCE} = -\sum_{j \in \mathbb{M}(x)} (y_j \log \sigma_\theta^j(x) + (1-y_j)\log(1-\sigma_\theta^j(x)))$$

$$\mathbb{M}(x) = \{j : j \in L \wedge l_j \overset{\mathrm{rel}}{\sim} x\}$$

where $l_j \overset{\mathrm{rel}}{\sim} x$ means label $l_j$ is relevant to $x$. By matching a category of $x$ with categories in the attribute taxonomy, we can obtain all possible attribute-value pairs for $x$. We regard those pairs as relevant labels to $x$.

By introducing the function $\mathbb{M}$, BCE loss discards the log loss values from the irrelevant labels. The label masking enables us to train XMC models more properly since (1) it reduces bias in the distribution between positive and negative labels, and (2) the irrelevant labels would not affect the model parameters during back-propagating. This makes the model training more sensitive than normal to misclassification within relevant labels.

## 5 Experiments

### 5.1 Dataset

We use product data and attribute taxonomy from Rakuten[2], a large e-commerce platform in Japan. Each product consists of a tuple of category, title, description and a set of attribute-value

[2] https://www.rakuten.co.jp/

| | Count |
|---|---|
| # of product data | 1,999,175 |
| # of top categories | 38 |
| # of leaf categories | 6,796 |
| # of distinct attributes | 1,300 |
| # of distinct attribute-value pairs (labels) | 7,979 |
| Avg. tokens per title | 44.05 |
| Avg. tokens per description | 332.04 |
| Avg. # of positive labels | 4.42 |
| Avg. # of negative labels | 7974.58 |
| Avg. # of relevant labels | 489.25 |
| Avg. # of irrelevant labels | 7489.75 |

Table 2: Data statistics. These numbers are calculated from both training and test data.

pairs. Rakuten manages category and attribute taxonomies, and sellers assign products a category and attribute-value pairs defined in the taxonomies. Figure 2 shows an example of the product data.

For experiments, among product data in Rakuten, we randomly sampled 2,000,446 product data that own one or more attribute-value pairs except brands. We halve this dataset as a 50-50 train/evaluation split. We selected attribute-value pairs appeared in both datasets[3], and removed product data that did not have any selected pairs. Moreover, from the evaluation dataset, we discarded product data whose category did not appear in the training dataset. As a result, the training and evaluation datasets contain 1,000,047 and 999,128 products respectively. Statistics of the dataset are listed in Table 2. We can see that the label masking reduces the size of labels from 7,979 to 489 on average.

### 5.2 Evaluation Metrics

We use precision (P), recall (R), $F_1$ score and precision at $k$ (P@k, $k$ = 1,3,5), which is widely used in the XMC tasks.To obtain a top-$k$ list, we regard all prediction results as output regardless of scores.

### 5.3 Models

We compare the following models:

**XR-Transformer** XMC model that shows the state-of-the-art performance on datasets commonly used in the XMC field (Zhang et al., 2021). We train the model using the codes released from the authors[4] with default parameters other than max sequence length and batch size. We set 512 for max sequence length and 64 for batch size.

| Hyper parameter | Value |
|---|---|
| Learning rate | 0.0001 |
| Weight decay | 0.0 |
| Epoch | 5 |
| Batch size | 64 |
| Dropout rate | 0.1 |
| Max. sequence length | 512 |
| Warmup proportion | 10% |
| # of CLS's hidden representations to concat. ($l$) | 5 |

Table 3: Hyper parameters

**BERT** BERT (Devlin et al., 2019) without our label masking. It computes BCE loss from all labels.

**BERT with multiple classifiers** Model that simply exploits a given category. We design a classifier (feed forward) layer for each category, and put them on the top of a single BERT. Because of this, parameters in BERT are in common with all classifiers. According to the category, we replace a classifier in training and testing. We construct mini-batches to include product data in the same category. As categories, in addition to leaf categories (*e.g., Sneakers*), we also adopt top categories (*Shoes*). This is because the size of training data is not sufficient in some minor leaf categories. By taking top categories, we can expect that the size of training data is enlarged although it increases irrelevant labels to leaf categories assigned in products. The total number of top categories is 38, including shoes, food, furniture and home appliances.

**BERT with label masking** Our proposed model. It computes BCE loss from only relevant attribute-value pairs to the category of given product data. Unlike BERT with multiple classifiers, this model has a single classifier, and the classifier is trained using product data from all categories.

For fair comparison with our model that assumes a category of the target product to be given, we discard irrelevant labels that the baseline models predict.

We employ a pretrained Japanese BERT-base model and its tokenzier released from Tohoku University[5], and use them in all models. We apply NFKC Unicode normalization[6] to titles and descriptions before the tokenization.

[3] The total number of the un-selected attribute-value pairs is 1,289. These pairs appeared 10 times or less in the sampled product data.

[4] https://github.com/amzn/pecos

[5] https://github.com/cl-tohoku/bert-japanese

[6] https://unicode.org/reports/tr15/

| Models | Micro |  |  | Macro |  |  | P@k (%) |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | P (%) | R (%) | $F_1$ | P (%) | R (%) | $F_1$ | k = 1 | k = 3 | k = 5 |
| XR-Transformer | **92.01** | 73.80 | 81.90 | 45.43 | 19.93 | 27.71 | 90.30 | 65.68 | 53.61 |
| BERT | 88.77 | 74.64 | 81.09 | 26.57 | 15.42 | 19.51 | 87.59 | 63.97 | 52.36 |
| BERT w/ multiple classifiers - leaf | 87.79 | 74.90 | 80.83 | 47.24 | 30.89 | 37.36 | 87.79 | 55.63 | 44.60 |
| BERT w/ multiple classifiers - top | 89.04 | 79.88 | 84.21 | 52.12 | 34.99 | 41.87 | 91.10 | 65.95 | 53.75 |
| BERT w/ label masking (ours) | 88.90 | **80.46** | **84.47** | **52.82** | **35.85** | **42.71** | **91.57** | **66.31** | **54.08** |

Table 4: Performance of each model.

| Group<br>(# of pairs) | Freq. | Micro $F_1$ |  | Macro $F_1$ |  |
|---|---|---|---|---|---|
| High (76) | $[10^4, \infty)$ | 89.57 | (+1.56) | 86.15 | (+2.31) |
| Med. (454) | $[10^3, 10^4)$ | 81.98 | (+3.90) | 78.58 | (+5.24) |
| Low (1,457) | $[10^2, 10^3)$ | 70.79 | (+10.59) | 66.80 | (+14.55) |
| Rare (5,992) | $[1, 10^2)$ | 53.85 | (+33.20) | 33.19 | (+26.97) |

Table 5: Micro and macro $F_1$ scores of our model for each group of attribute-value pairs. Gains over BERT without label masking are enclosed in parentheses.

For models other than XR-Transformer, we use gradient descent by the Adam (Kingma and Ba, 2015) optimizer. To avoid overfitting, we apply a dropout rate at 0.1 and stochastic weight averaging (Izmailov et al., 2018) to the models. Table 3 shows the hyper parameters.

Similarly with our model, as the representation of the input to BERT and BERT with multiple classifiers, we use a vector concatenating CLS embeddings obtained from the last five encoders. We implemented the models in PyTorch.

### 5.4 Results

Table 4 shows the performance of each model. We can observe that our proposed model outperformed all baselines. Micro and macro $F_1$ gains over BERT without label masking are 3.38 and 23.20 points, respectively. The significant improvement on macro $F_1$ score shows that the label masking is effective on various kinds of attribute-value pairs. These results show that reducing the number of irrelevant labels in training is crucial to train more accurate XMC models.

The reason why the performance of BERT with multiple classifiers trained on leaf categories is lower than ours is that the number of training examples for this model is insufficient in many leaf categories, as we mentioned. For 5,572 categories, the number of training examples is less than 64. Since parameters of BERT in this model are in common with all categories, this result implies that the classifiers are not well trained. On the other hand, the single classifier in our model is successfully trained because (general) attribute-value pairs scattered on various leaf categories are fully used to train the classifier.

Since the data sparseness problem is alleviated, BERT with multiple classifiers trained on top categories outperforms the model trained on leaf categories. Furthermore, its performance is closed to ours. We believe that the gap of the performance between the model trained on top categories and ours is from the quality of association between categories and attributes. In case of the model trained on top categories, attribute-value pairs defined for different leaf categories in the same top category are handled as relevant labels (*e.g., heel height for sneakers*). Meanwhile, our model is not affected by such attribute-value pairs. The gap implies that these erroneous relevant pairs hurt the performance.

To see the effectiveness of the label masking in detail, we categorize attribute-value pairs according to the frequency in the training data, and then check the performance for each frequency group. Table 5 shows the performance of our model in each group together with micro and macro $F_1$ gains over BERT. The improvement in micro and macro $F_1$ scores is greater for attribute-value pairs with less training examples. This means that the label masking works well for attribute-value pairs at the long-tail.

## 6 Conclusion

In this paper, we formalized AVE as XMC, and proposed label masking, a simple and effective method that mitigates the negative impact from the imbalanced distribution of attribute-value pairs relevant and irrelevant to products. Experimental results using a real-world dataset show that the label masking improves the performance of BERT-based XMC models; it is especially effective for attributes with less training data.

As for future work, we plan to see the effectiveness of the label masking method on other tasks in e-commerce domains such as item classification.

## Acknowledgement

We thank Naoki Yoshinaga for the fruitful comments before the submission. We also thank the anonymous reviewers for their careful reading of our paper and insightful comments.

## References

Lidong Bing, Tak-Lam Wong, and Wai Lam. 2012. Unsupervised extraction of popular product attributes from web sites. In *Information Retrieval Technology, 8th Asia Information Retrieval Societies Conference, AIRS 2012*, volume 7675 of *Lecture Notes in Computer Science*, pages 437–446, Berlin, Heidelberg. Springer.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. *Taming Pretrained Transformers for Extreme Multi-Label Text Classification*, KDD '20, page 3163–3171. Association for Computing Machinery, New York, NY, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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*, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI).

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7987–7994.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8489–8502, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the third International Conference on Learning Representations*, San Diego, California, USA.

Mikko I. Malinen and Pasi Fränti. 2014. Balanced k-means for clustering. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 32–41, Berlin, Heidelberg. Springer Berlin Heidelberg.

A. Mittal, K. Dahiya, S. Agrawal, D. Saini, S. Agarwal, P. Kar, and M. Varma. 2021a. Decaf: Deep extreme classification with label features. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, WSDM '21, page 49–57, New York, NY, USA. Association for Computing Machinery.

A. Mittal, N. Sachdeva, S. Agrawal, S. Agarwal, P. Kar, and M. Varma. 2021b. Eclare: Extreme classification with label graph correlations. In *Proceedings of The ACM International World Wide Web Conference*, WWW '21, page 3721–3732, New York, NY, USA. Association for Computing Machinery.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. In *KDD 2016 Workshop on Enterprise Intelligence*, San Francisco, California, USA.

Katharina Probst, Rayid Ghani, Marko Krema, Andrew E. Fano, and Yan Liu. 2007. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2838–2843, Hyderabad, India. Morgan Kaufmann Publishers Inc.

Duangmanee Putthividhya and Junling Hu. 2011. 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. Association for Computational Linguistics.

Martin Rezk, Laura Alonso Alemany, Lasguido Nio, and Ted Zhang. 2019. Accurate product attribute extraction on the field. In *Proceedings of the 35th IEEE International Conference on Data Engineering*, pages 1862–1873, Macau SAR, China. IEEE.

Keiji Shinzato and Satoshi Sekine. 2013. Unsupervised extraction of attributes and their values from product description. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1339–1347, Nagoya, Japan. Asian Federation of Natural Language Processing.

Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2022. Simple and effective knowledge-driven query expansion for QA-based product attribute extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. (to appear).

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to extract attribute value from product via question answering: A multi-task approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, pages 47–55, New York, NY, USA. Association for Computing Machinery.

Tak-Lam Wong, Wai Lam, and Tik-Shun Wong. 2008. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 35–42, New York, NY, USA. Association for Computing Machinery.

Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczyński. 2018. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6358–6368, Red Hook, NY, USA. Curran Associates Inc.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hanchu Zhang, Leonhard Hennig, Christoph Alt, Changjian Hu, Yao Meng, and Chao Wang. 2020. Bootstrapping named entity recognition in E-commerce with positive unlabeled learning. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 1–6, Seattle, WA, USA. Association for Computational Linguistics.

Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 34, pages 7267–7280. Curran Associates, Inc.

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