Does Named Entity Recognition Truly Not Scale up to Real-world Product Attribute Extraction?

Wei-Te Chen[†] **Keiji Shinzato**[†] **Naoki Yoshinaga**[‡] **Yandi Xia**[†]

† Rakuten Institute of Technology, Rakuten Group, Inc.
‡ Institute of Industrial Science, The University of Tokyo
† {weite.chen, keiji.shinzato, yandi.xia}@rakuten.com
‡ {ynaga}@iis.u-tokyo.ac.jp

Abstract

The key challenge in the attribute-value extraction (AVE) task from e-commerce sites is the scalability to diverse attributes for a large number of products in real-world e-commerce sites. To make AVE scalable to diverse attributes, recent researchers adopted a question-answering (QA)-based approach that *additionally* inputs the target attribute as a query to extract its values, and confirmed its advantage over a classical approach based on named-entity recognition (NER) on real-word e-commerce datasets. In this study, we argue the scalability of the NER-based approach compared to the QA-based approach, since researchers have compared BERT-based QA-based models to only a weak BiLSTM-based NER baseline trained from scratch in terms of only accuracy on datasets designed to evaluate the QA-based approach. Experimental results using a publicly available real-word dataset revealed that, under a fair setting, BERT-based NER models rival BERT-based QA models in terms of the accuracy, and their inference is faster than the QA model that processes the same product text several times to handle multiple target attributes.

1 Introduction

To serve better product search and recommendation to customers on e-commerce sites, industry researchers have studied attribute value extraction (AVE) to organize hundreads of millions of products in terms of their attribute values. In the literature, AVE has been formalized as sequence tagging similar to named entity recognition (NER), which recognizes attribute values in the given product text while classifying them to corresponding attributes defined by an e-commerce site (Figure 1) (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). When we apply NER-based sequence tagging to AVE, a larger number of classes (attributes), which can exceed a thousand, poses a challenge.

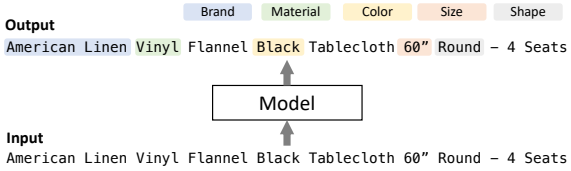

Figure 1: Overview of attribute value extraction.

To make AVE scalable to thousands of attributes, Xu et al. (2019) have proposed models based on question-answering (QA) to reduce the number of classes by additionally inputting the target attribute and extract only values for that attribute. They reported that the NER-based model, OpenTag (Zheng et al., 2018) performed poorly on rare attributes due to a data sparseness problem and thus did not scale to the diverse attributes. Following this study, recent researchers focus on the QA-based approach (Wang et al., 2020; Yang et al., 2022; Shinzato et al., 2022).

In this study, we re-evaluate the scalability of the NER-based approach to real-world AVE against the QA-based approach in a more fair setting, in terms of efficiency in inference as well as accuracy. In the above comparison (Xu et al., 2019), OpenTag is based on a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) trained from scratch (Figure 2 (a)), whereas the QA-based approaches leverage a pre-trained BERT, which remedies the data sparseness problem. Meanwhile, the NER-based approaches may classify recognized values as irrelevant attributes and have issues in recognizing overlapping values (Shinzato et al., 2023), whereas the QA-based approaches bypass these issues by explicitly giving a single target attribute. We should also consider the scalability to a number of products on e-commerce sites, since the AVE model will be applied to hundreds of millions of product text on major e-commerce sites. The NER-based

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 152–159

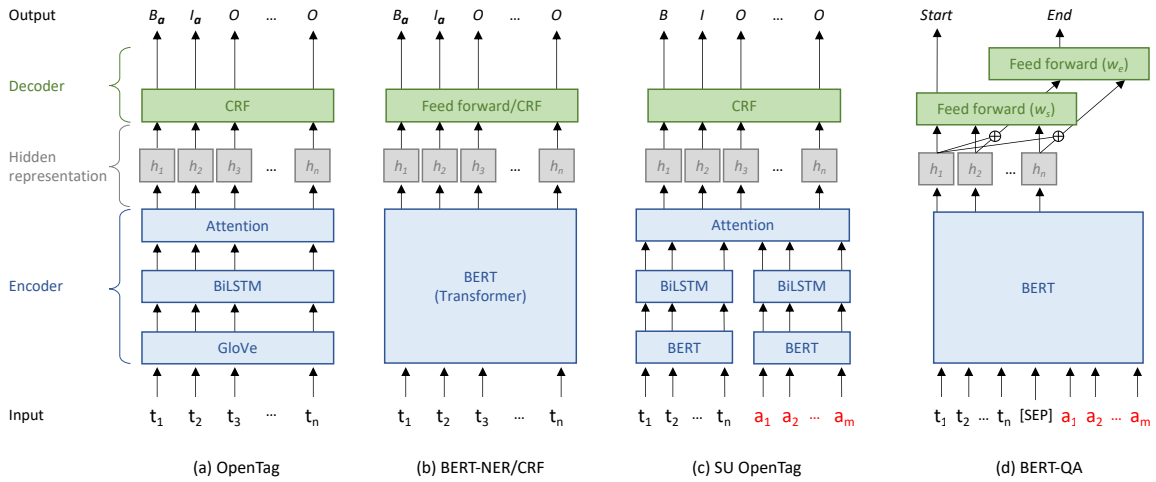

Figure 2: Comparison of model architectures for the AVE task. The common input to the models is text $\boldsymbol{t} = \{t_1, \ldots, t_n\}$. $\oplus$ represents the concatenation operation. OpenTag (Zheng et al., 2018) and BERT-NER/CRF (Devlin et al., 2019; Yan et al., 2021) introduce a set of chunk tags for each attribute (*e.g.,* $B_{\boldsymbol{a}}$). Meanwhile, SU OpenTag (Xu et al., 2019) and BERT-QA (Wang et al., 2020) take a target attribute $\boldsymbol{a} = \{a_1, \ldots, a_m\}$ as an additional input; SU OpenTag and BERT-QA thereby predicts a single set of chunk tags and starting and ending positions, respectively, to extract a value corresponding to the given attribute. In our experiments, to enable BERT-QA to extract multiple values for a given attribute, we replace a decoder part in the model with a feed forward layer used in (b).

models can extract values for multiple attributes at once, whereas the QA-based models can extract values for only a single target attribute at once and require multiple runs if the input text includes values for more than one attribute.

We evaluate BERT-NER (Devlin et al., 2019) models (Figure 2 (b)) on a publicly available real-world AVE dataset (Yang et al., 2022), and confirm that BERT-NER scales up to a thousand of attributes in terms of accuracy, with a smaller inference cost. In fact, the BERT-based NER model performs as well as the BERT-based QA model when it does not predict irrelevant attributes and the input does not include overlapping values.

Our contribution is as follows.

- We evaluated BERT-based NER models for AVE using a publicly available real-world dataset for the first time.
- We found that the QA models are superior to the NER models in that they i) can handle overlapping values for multiple attributes and ii) can avoid predicting wrong attributes thanks to their formulation that explicitly inputs the target attribute for extraction.
- We confirmed the BERT-based NER models require a smaller inference cost against the QA-based models, thus showing better scalability to the number of products.

2 Related Work

Traditionally, most previous studies formulated AVE as a sequence tagging problem and adapted NER techniques (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; Zhu et al., 2020; Yan et al., 2021). These studies introduce a set of chunk tags (*e.g.,* BIO tags) for each attribute and classify each token in text into one of the chunk tags. Therefore, NER-based models can extract values for multiple attributes at the same time. However, since the number of attributes in real-world AVE easily exceeds a thousand (Xu et al., 2019), the models are required to perform a large-scale multi-class classification at the token level.

To address a large number of attributes in the AVE task, recent studies (Xu et al., 2019; Wang et al., 2020; Yang et al., 2022; Shinzato et al., 2022) adopted a QA-based approach (*e.g.,* Figures 2 (c) and (d)). 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. By taking attributes as the input, QA-based models achieved the best performance on publicly available AVE datasets (Wang et al., 2020; Yang et al., 2022). On the other hand, unlike NER-based models, QA-based models cannot

extract values for multiple attributes at the same time. This is because the models jointly encode a given title and attribute, and it is necessary to perform extraction multiple times when there are values for multiple attributes in the title. Hence, the QA-based models are more time-consuming than NER-based models, which incurs a critical issue in business contexts.

Previous studies (Xu et al., 2019; Wang et al., 2020; Yang et al., 2022) reported that NER-based models did not scale up to large-sized attributes in AVE through the evaluation of OpenTag (Zheng et al., 2018), which was referred to as the state-of-the-art NER-based model. However, since OpenTag relies on bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) and GloVe (Pennington et al., 2014), it is debatable whether NER-based models are truly unscalable, as large-scale pre-trained language models such as BERT (Devlin et al., 2019) have become the de-facto standard as a text encoder. Although Yan et al. (2021) verified the performance of BERT-based NER models using their own dataset consisting of 12 attributes, the size of attributes is far from the AVE task in the real-world scenario. This paper is the first work that reports the performance of BERT-based NER models on a publicly available real-world dataset for AVE.

3 Attribute Value Extraction

We formalize AVE as a sequence labeling problem. Let $\boldsymbol{A}$ be the set of all possible attributes in training data and $\boldsymbol{Y}$ be the tag set containing all the tags. If we choose BIO as our chunk tag scheme, then $\boldsymbol{Y} = \{\{\boldsymbol{A} \times \{B, I\}\} \cup O\}$. Given a product data (text) $\boldsymbol{t} = \{t_1, t_2, \ldots, t_n\}$ where n is the number of tokens in $\boldsymbol{t}$, the model is trained to return $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ where $y_i \in \boldsymbol{Y}$. In short, the model performs multiclass classificaition over each token. In the inference, attributes and values are decoded from the sequence of predicted tags.

In what follows, we describe BERT (Devlin et al., 2019) and BERT-based NER models.

3.1 Preliminary: BERT

BERT is a large-scale language model based on Transformer (Vaswani et al., 2017). It is pre-trained with a large-scale text corpus following masked-language modeling (MLM) and next-sentence prediction (NSP). MLM learns the semantics of each word from the surroundings, while the NSP learns the relation between text segment pairs.

BERT can be fine-tuned for downstream tasks such as sentiment classification and NER. In general, a task-specific layer is placed on top of BERT and is trained using labeled data for downstream tasks. Even if the size of the labeled data is small, BERT performs better because of pre-training with large-scale data. Thus, BERT has achieved great success as a text encoder in various NLP tasks.

3.2 NER with BERT

BERT-NER (Devlin et al., 2019) is composed of BERT followed by a sequence tagging layer (Figure 2(b)). BERT accepts a sequence of tokens $\boldsymbol{t}$ as input, and then encodes it into a list of contextualized dense vectors $\boldsymbol{h}$, each representing one token. Next, a sequence tagging layer classifies token t_i into a possible tag $y \in \boldsymbol{Y}$ following dense vector h_i. As a sequence tagging layer, we can use a feed-forward layer followed by a softmax layer received h_i. The probability of y given t_i is calculated as:

$$P(y|t_i) = \frac{\exp(\sigma(y, h_i))}{\sum_{y' \in \boldsymbol{Y}} \exp(\sigma(y', h_i))}$$

where σ is a learnable scoring function to estimate the score that the dense vector h_i and target y appear together.

However, the sequence tagging based on the feed-forward layer fails to classify tokens because it cannot capture the association between the neighborhood labels. To better consider the label association, a conditional random field (CRF) (Lafferty et al., 2001) layer is placed on top of BERT instead of the feed-forward layer. A linear-chain CRF layer considers the omission and transition scores simultaneously with the following probability formula:

$$P(\boldsymbol{y}|\boldsymbol{t}) = \frac{\exp(\sum_{i=1}^{n} \sigma(y_i, h_i) + \sum_{i=1}^{n-1} \tau(y_i, y_{i+1}))}{Z(\boldsymbol{t})}$$

where σ and τ are the learnable omission and transition scoring functions. $Z(\boldsymbol{t})$ is a partition term to normalize the probability distribution. The τ function estimated the score of transiting from the label y_i to the next label y_{i+1}. Thus, the τ function easily causes memory exhaustion when the label size is large; it requires $O(|\boldsymbol{Y}|^2)$ memory space.

In this paper, however, we did not use a CRF layer and adopted a simple BERT-NER for evaluation. This is because in addition to the memory space, Yan et al. (2021) reported that replacing the feed-forward layer with the CRF layer showed slightly poor performance in the AVE task.

	Train	Dev.	Test	Test_{NER}
Number of product data	640,000	100,000	290,773	248,493
Number of attribute value annotations	2,294,309	358,773	1,039,286	806,021
Number of attribute value annotations w/o NONE	1,901,226	297,527	862,308	650,992
Number of unique attributes	693	660	685	670
Number of unique values	54,200	21,734	37,092	30,178
Number of unique attribute-value pairs	63,715	25,675	43,605	34,953

Table 1: Statistics of the MAVE dataset. We randomly selected 640,000 examples from the entire training data following (Yang et al., 2022).

4 Experiments

We evaluate BERT-based NER models on a publicly available real-world AVE dataset, namely the MAVE dataset (Yang et al., 2022).[1] The statistics of the dataset are listed in Table 1.

Similar to (Yang et al., 2022), we verify the models on the following setups.

All attributes To evaluate the capability of scaling up to large-sized attributes, we evaluate the models on all attributes in the dataset.

Selected attributes To demonstrate the performance on individual attributes, we evaluate the models on a set of selected attributes; five head attributes that contain a large number of attribute-value annotations in the dataset and five tail attributes that have very few examples in the dataset. Those attributes were selected by Yang et al. (2022).

4.1 Dataset

The MAVE dataset (Yang et al., 2022) is composed of a curated set of 2.2M products from Amazon Review Data (Ni et al., 2019). The dataset contains various kinds of products such as shoes, clothing, watches, books, and home decor decals. Table 2 shows an example of product data in MAVE. As you can see, the textual data of the product consists of multiple sources. We simply concatenate all of them using a [SEP] token as a delimiter and regard the resulting text as an input to models.

The product data provide spans for each value and NONE for attributes if the values are not mentioned. Yang et al. (2022) employed the AVEQA model (Wang et al., 2020) and heuristic rules to obtain those spans and NONE. We straightforwardly use beginning and ending positions in each span to annotate values in the text for the experiments on all attributes. On the other hand, for experiments on the selected attributes, we only annotate the values of the selected attributes.

Source	Text
Title	**Wireless** Mobile Mouse 1000 - Maus - 3 Taste(n)
Description	Microsoft **Wireless** Mobile Mouse 1000 - MAGENTA PINK
Feature 1	9.09
Feature 2	18.18
Brand	Microsoft
Attribute	**Value**
Connectivity	{Wireless, Title, Span(0, 8)}, {Wireless, Description, Span(10, 18)}
Sensitivity	NONE

Table 2: Example product data in the MAVE dataset.

There are issues in using the MAVE dataset to evaluate the NER-based AVE models. First, the datasets provide a few target attributes for each example to evaluate the QA-based models. Since NER-based models do not utilize these attributes, they may recognize values as attributes other than the target attributes. Moreover, the datasets include redundant overlapping attributes (Table 4 in (Shinzato et al., 2023)), which require nested NER (Wang et al., 2022) to handle by the NER-based approach. Note that the QA-based approach unfairly bypasses these issues by explicitly giving one target attribute for extraction.

To see the impact of these issues, we use not only the original MAVE test set but also its subset (Test_{NER}) in which i) the BERT-NER model did not predict attributes other than the target attributes (19,278 examples) and ii) examples do not include overlapping values for multiple attributes (24,042 examples). As the training set, similarly to (Yang et al., 2022), we randomly selected 640,000 product data from the original 2.2M training data to make the training faster. As the development set, we randomly selected another 100,000 product data from the original training data.

[1] https://github.com/google-research-datasets/MAVE

Attributes	Test						Test_{NER}					
	BERT-NER			BERT-QA			BERT-NER			BERT-QA		
	P(%)	R(%)	F_1	P(%)	R(%)	F_1	P(%)	R(%)	F_1	P(%)	R(%)	F_1
(*All Attributes*)	96.35	83.22	89.30	95.39	91.74	93.53	96.48	89.49	92.85	95.44	92.51	93.95
(*Selected Attributes - Head*)												
Type	95.89	90.03	92.87	95.42	91.77	93.56	95.89	91.20	93.49	95.40	92.48	93.92
Style	96.48	88.55	92.34	96.32	92.60	94.42	96.72	90.73	93.63	96.33	93.24	94.76
Material	96.50	87.56	91.82	95.54	93.23	94.37	96.62	89.27	92.80	95.76	94.06	94.90
Size	93.79	76.32	84.16	91.18	90.76	90.97	94.24	78.64	85.74	91.38	91.17	91.27
Capacity	96.96	87.48	91.98	95.44	93.41	94.41	96.24	86.06	90.87	94.65	92.49	93.56
(*Selected Attributes - Tail*)												
Black Tea Variety	100.00	25.71	40.91	87.88	82.86	85.29	*No extraction results*			62.50	100.00	76.92
Staple Type	98.08	77.27	86.44	96.72	89.39	92.91	100.00	79.66	88.68	100.00	93.22	96.49
Web Pattern	95.45	70.00	80.77	100.00	93.33	96.55	95.45	70.00	80.77	100.00	93.33	96.55
Cabinet Configuration	100.00	68.29	81.16	97.50	95.12	96.30	100.00	62.07	76.60	96.43	93.10	94.74
Power Consumption	92.11	77.78	84.34	97.56	88.89	93.02	90.91	88.24	89.55	96.97	94.12	95.52

Table 3: Performance of models on all and selected attributes in MAVE. Average refers to averaged performance on the selected attributes. The number of parameters in BERT-NER and BERT-QA is 110M and 108M, respectively.

4.2 Models

We compare the following models:

BERT-NER NER-based model used in (Devlin et al., 2019) (Figure 2 (b)). It utilizes a feed-forward layer to decode hidden representations to tags.

BERT-QA QA-based model proposed in (Wang et al., 2020) (Figure 2 (d)). It jointly encodes a given text and attribute by feeding a string concatenating them to BERT. Then, it computes the probabilities for the start index s and the end index e of the value span for the given attribute.

$$s = \arg\max_i(\mathsf{softmax}(w_s h_i))$$
$$e = \arg\max_{i \geq s}(\mathsf{softmax}(w_e(\mathsf{Concat}(h_i, h_s))))$$

where h_i is hidden representation of the *i-th* token in the given text. w_s and w_e are two matrices that map the hidden representations to the output logits for the start and end indices, respectively. By concatenating h_i and h_s, the model incorporates the begin-end dependency. Since this decoding method cannot extract multiple values for a given attribute, we replace it with a feed-forward layer that we use in BERT-NER.

For all models, we adopt BILOU (Sekine et al., 1998; Ratinov and Roth, 2009) as a chunk tag scheme. Therefore, the total number of labels is $N \times 4 + 1$, where N is the number of distinct attributes in the training data in the case of BERT-NER whereas $N = 1$ in the case of BERT-QA.

4.3 Evaluation Measure

Following the literature (Yang et al., 2022), we used micro precision (P), recall (R), and F_1 score as evaluation metrics, and computed those metrics by span basis. In the MAVE dataset, there are attributes whose values do not appear in the given text (negative). For the ground truth with such no attribute values, models can predict no values, or incorrect values (FP_n) while for the ground truth with concrete attribute values, the model can predict no values (FN), correct values (TP), or incorrect values (FP_p). Based on those types of predicted values, P and R are computed as follows:

$$\text{P} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}_\text{p}| + |\text{FP}_\text{n}|}, \text{R} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|}$$

F_1 is computed as $2\times$ P $\times$ R / (P + R).

As mentioned in Section 4.2, while BERT-QA refers to an attribute as input, the NER-based models do not. To fairly compare the NER-based models with BERT-QA, we discard extracted values if there are no ground truth labels for the attributes for evaluation with all test examples.

4.4 Implementation Details

We implemented all models in PyTorch (Paszke et al., 2019) in ver. 1.11.0. For the underlying BERT pre-trained model, we used the "bert-base-cased"[2] in transformers (Wolf et al., 2020). We used 2 NVIDIA 80 GB A100 GPUs in all experiments. In training, we used Adam (Kingma and

[2] https://huggingface.co/bert-base-cased

Ba, 2015) as the optimizer for all the models. The learning rate is set to 5×10^{-5} for all BERT-based models. We trained models up to 20 epochs with a batch size of 32. We selected the best model according to micro F_1 on the dev set.

4.5 Results

Accuracy

Table 3 shows the experimental results on all and selected attributes in the MAVE dataset.

All attributes Similarly to (Wang et al., 2020), BERT-QA shows better performance than NER-based models for all test examples in our experiments. However, the BERT-NER model exhibits comparable accuracy to the BERT-QA model on test_{NER} where the model does not predict attributes that are not included in the target attributes and test examples do not include overlapping attributes. Thus, the advantage of the BERT-QA model is basically obtained by supporting overlapping values for multiple attributes and by avoiding generating irrelevant attributes by giving a target attribute.[3]

Selected attributes Similarly to the results on all attributes, the BERT-QA model outperforms the BERT-NER model for all text examples. Again, this gain was reduced when the models are evaluated on test_{NER}.

Inference time

Table 4 shows the inference time of the BERT-NER and BERT-QA models on all test examples. The BERT-NER model is faster than the BERT-QA model because the QA-based model must be applied to the same product text multiple times varying input attributes of interest. Meanwhile, NER-based models perform only once regardless of the number of attributes. This performance gap becomes larger when we apply the models to product text that contains more attributes (Shinzato et al., 2023) or when the taxonomy cannot narrow down the target attribute.

To make QA-based models accurate and efficient, it is a must to prepare a comprehensive attribute taxonomy to cover necessary and sufficient attributes for the target product (text) to avoid the wrong extraction of irrelevant attributes and to minimize the number of runs on the same inputs and not to extract irrelevant attributes. If such a taxonomy is not available, we need to run the QA-based model with all possible attributes. It will result in a long inference time as well as the extraction of irrelevant attributes. In light of the above, the BERT-NER model, which works without using a comprehensive taxonomy, could be a robust and practical solution to AVE; researchers should revisit the NER-based approach as an important research target in AVE.

[3]The accuracy gain of BERT-QA models was attributed mostly to supporting extraction of overlapping values; The P/R/F_1 of the BERT-NER model was $96.28/89.14/92.58$ for test examples without overlapping values, while those of the BERT-QA model was $95.44/92.35/93.87$.

Model	Time (sec)
BERT-NER	905
BERT-QA	1,464

Table 4: Inference time on Test.

5 Conclusions

In this study, we have revisited the NER-based approach to attribute-value extraction (AVE) from e-commerce sites, and evaluated the scalability of BERT-based NER models on the AVE task. We performed experiments using a publicly available real-world dataset and confirmed that even NER-based models scaled up to large-sized attributes. These results showed that experiments with OpenTag are insufficient to verify the scalability and performance of NER-based models in real-world AVE.

We observed that the BERT-based NER model rivals the BERT-based QA model in terms of accuracy for test examples in which the model does not predict attributes other than the target attributes and examples do not include overlapping values for multiple attributes; these issues are bypassed in the QA-based approach by explicitly giving a single target attribute for extraction. Even in the NER-based approach, limiting the output tag space to the target attributes will remedy the first issue, and the use of nested NER models (surveyed in (Wang et al., 2022)) will remedy the second issue.

Moreover, QA models are more time-consuming since the models must be applied to the same product text for each target attribute (for thousands of attributes when a comprehensive attribute taxonomy narrows down the candidates). We thus conclude that the NER-based models still can be a practical solution to AVE, and worth a target for future research.

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.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., Red Hook, NY, USA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. 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.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Sixth Workshop on Very Large Corpora*, pages 171–178, Quebec, Canada.

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 (Volume 2: Short Papers)*, pages 227–234, Dublin, Ireland. Association for Computational Linguistics.

Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2023. A unified generative approach to product attribute-value identification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6599–6612, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Red Hook, NY, USA. Curran Associates, Inc.

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.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data*, 16(6).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 38–45, Online. Association for Computational Linguistics.

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.

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.

Jun Yan, Nasser Zalmout, Yan Liang, Christan Grant, Xiang Ren, and Xin Luna Dong. 2021. AdaTag: Multi-attribute value extraction from product profiles with adaptive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4694–4705, Online. Association for Computational Linguistics.

Li Yang, Qifan Wang, Zac Yu, Anand Kulkarni, Sumit Sanghai, Bin Shu, Jon Elsas, and Bhargav Kanagal. 2022. MAVE: A product dataset for multi-source attribute value extraction. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1256–1265. ACM.

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.

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.

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal joint attribute prediction and value extraction for E-commerce product. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2129–2139, Online. Association for Computational Linguistics.