CoMave: Contrastive Pre-training with Multi-scale Masking for Attribute Value Extraction

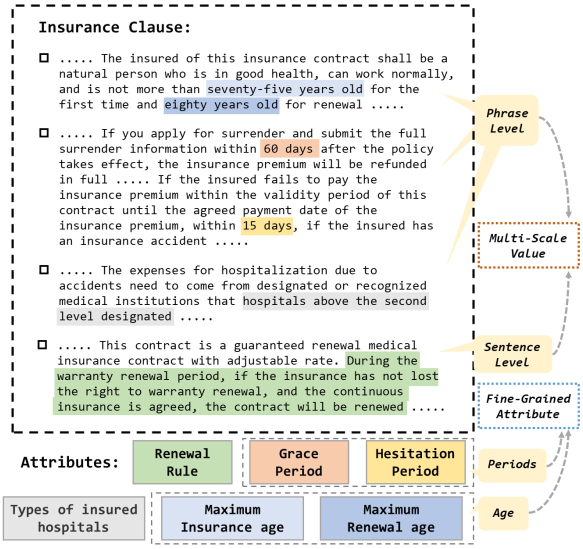
Xinnan Guo[1], Wentao Deng[2], Yongrui Chen[1], Yang Li[1], Mengdi Zhou[2], Guilin Qi[1], Tianxing Wu[1], Yang Dong[2], Liubin Wang[2], Yong Pan[2]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Ant Group, China
guoxinnan0727@163.com, dengwentao362502@icloud.com, yrchen@seu.edu.cn,
ly200170@alibaba-inc.com, jacquelyn.zmd@antgroup.com,
gqi, tianxingwu@seu.edu.cn, dongyang1231@gmail.com

Abstract

Attribute Value Extraction (AVE) aims to automatically obtain attribute value pairs from product descriptions to aid e-commerce. Despite the progressive performance of existing approaches in e-commerce platforms, they still suffer from two challenges: 1) difficulty in identifying values at different scales simultaneously; 2) easy confusion by some highly similar fine-grained attributes. This paper proposes a pre-training technique for AVE to address these issues. In particular, we first improve the conventional token-level masking strategy, guiding the language model to understand multi-scale values by recovering spans at the phrase and sentence level. Second, we apply clustering to build a challenging negative set for each example and design a pre-training objective based on contrastive learning to force the model to discriminate similar attributes. Comprehensive experiments show that our solution provides a significant improvement over traditional pre-trained models in the AVE task, and achieves state-of-the-art on four benchmarks[1].

1 Introduction

Product features are crucial components of e-commerce platforms and are widely used in applications such as product recommendation (Cao et al., 2018), product retrieval (Magnani et al., 2019), and product question answering (Yih et al., 2015; Chen et al., 2021b). Each product feature typically consists of an *attribute* and one or more *values*, providing detailed product descriptions to help customers make purchasing decisions. In recent years, Attribute Value Extraction (AVE) (Xu et al., 2019; Zhu et al., 2020; Yan et al., 2021) methods have received increasing attention because they can automatically extract product features from a massive amount of unstructured product text, with impressive results in e-commerce platforms, such as Amazon, AliExpress, and JD.



Insurance Clause:

□ ..... The insured of this insurance contract shall be a natural person who is in good health, can work normally, and is not more than seventy-five years old for the first time and eighty years old for renewal .....

□ ..... If you apply for surrender and submit the full surrender information within 60 days after the policy takes effect, the insurance premium will be refunded in full ..... If the insured fails to pay the insurance premium within the validity period of this contract until the agreed payment date of the insurance premium, within 15 days, if the insured has an insurance accident .....

□ ..... The expenses for hospitalization due to accidents need to come from designated or recognized medical institutions that hospitals above the second level designated .....

□ ..... This contract is a guaranteed renewal medical insurance contract with adjustable rate. During the warranty renewal period, if the insurance has not lost the right to warranty renewal, and the continuous insurance is agreed, the contract will be renewed .....

Figure 1: An example of attribute value extraction in the insurance field. Wherein each insurance clause contains multi-scale values and fine-grained similar attributes.

However, as e-commerce grows, some emerging domains, such as finance, insurance, and healthcare, bring two new challenges: a) **Multi-scale values**. Unlike normal products (e.g., clothing) with only short values (e.g., *color*: *red*), insurance products can have a value of a longer phrase or even multiple sentences. For example, the value of attribute *renewal rule* in Figure 1 contains more than 25 words (in green), rendering it impractical to retrieve them using related techniques such as Name Entity Recognition (NER) (Li et al., 2020; Yang et al., 2021). b) **Fine-grained divisions of attributes**. Compared with the coarse division of attributes in traditional e-commerce (e.g., *color*, *size*, and *material*), the division in insurance products is more refined, resulting in different attributes often having similar types. For instance, in the insurance clauses in Figure 1, *maximum insurance age* and *maximum renewal age* are both ages, and *grace period* and *hesitation period* are both periods. This fine-grained division makes the distinction be-

[1] https://github.com/ygxw0909/CoMave

Findings of the Association for Computational Linguistics: ACL 2023, pages 6007–6018
July 9-14, 2023 ©2023 Association for Computational Linguistics

tween the different attributes subtle, thus increasing the difficulty to distinguish between them.

Although recent pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) achieve tremendous success on a spectrum of NLP tasks, including AVE, we argue that they are not sufficient for the challenges above. First, the conventional Masking Language Model (MLM) focuses on token-level recovery and does not consider multi-scale values. Second, there is still a gap between the unsupervised general objectives and the downstream AVE in terms of task form, such that the model cannot benefit from pre-training when retrieving attributes, let alone distinguishing between fine-grained similar attributes..

In this paper, we propose CoMave, a novel PLM for AVE tasks. Relying on the large-scale corpus of triples $\langle \textit{text}, \textit{attribute}, \textit{value} \rangle$ collected by distant supervision, we propose three pre-training objectives to address the challenges: a) **Multi-Scale Masked Language Model** (MSMLM). We extend token-level recovery to the phrase as well as the sentence level, using different masking mechanisms to force the model to perceive spans of various lengths, thus providing a basis for identifying values at different scales. b) **Contrastive Attribute Retrieval** (CAR). To adapt the model to the fine-grained division of attributes, we require it to retrieve the correct attributes from a challenging candidate set of semantically similar attributes. The candidates are mainly collected by clustering and a contrastive loss is designed to help the model perceive the subtle differences between them. c) **Value Detection** (VD). To close the gap between pre-training and downstream AVE and further enhance the model's perception of values extraction, we let the model recognize all values without considering the corresponding attribute. To fully evaluate our pre-trained CoMave, we construct a new challenging benchmark INS. It consists of financial and medical texts from real scenarios and the corresponding manual annotations and is full of the two challenges we mentioned. Comprehensive experiments on four AVE datasets including INS demonstrate that, equipped with only a simple fine-tuning output layer, our CoMave not only achieves state-of-the-art results on the hardest INS but also outperforms all the compared methods on existing benchmarks.

Our contributions are summarized as follows:

- We release an advanced pre-trained language model, namely CoMave, for solving common challenges in AVE tasks. To the best of our knowledge, this is the first pre-training model aimed at AVE tasks.
- We propose three novel pre-training objectives: Multi-Scale MLM allows the model to adapt to values span of different scales, CAR uses contrastive loss to force the model to perceive subtle differences in similar attributes, and VD bridges the gap between pre-training and downstream tasks.
- Our method obtains state-of-the-art results on four AVE benchmarks, achieving significant improvements compared to existing PLMs.

2 Preliminaries

Given a natural language text $\mathcal{T}$ and a set of candidate attributes set $\mathcal{A} = \{a_1, a_2, ..., a_{|\mathcal{A}|}\}$, where a_i is an attribute, the goal of AVE is to extract a set $\mathcal{Y} = \{(a_1^*, \mathcal{V}_1), ..., (a_n^*, \mathcal{V}_n)\}$, where $a_i^* \in \mathcal{A}$ and $\mathcal{V}_i$ is the set of values belonging to a_i^*. For simplicity, each value $v \in \mathcal{V}_i$ is defined as a span of $\mathcal{T}$. In general, $\mathcal{T}$ is collected from a large number of product-related documents or other data sources, and $\mathcal{A}$ is a collection of attributes for various products in different categories.

Note that although formally AVE is similar to NER, the two still have significant differences, as we mentioned in section 1. First, the division of attributes is more fine-grained than the division of entity types (e.g., *location* and *person*). Second, the scale of entities is generally shorter, while that of values varies from token level to sentence level. Therefore, conventional NER methods are difficult to directly port to AVE tasks.

3 Methodology

3.1 Pre-training Corpus Construction

The pre-training procedure of CoMave requires a large-scale corpus $\mathcal{C} = \{(\mathcal{T}_i, \mathcal{A}_i, \mathcal{Y}_i)\}_M$ containing tens of millions of data. Manual annotation of such a large corpus is obviously impractical, thus we designed an automatic method to construct $\mathcal{C}$. In brief, we first collect the triples $\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle$ from several existing open-domain knowledge graphs, including DBpedia (Lehmann et al., 2015), Yago (Tanon et al., 2020), WikiData (Vrandecic and Krötzsch, 2014), and OpenKG (Chen et al., 2021a). Then, we regard

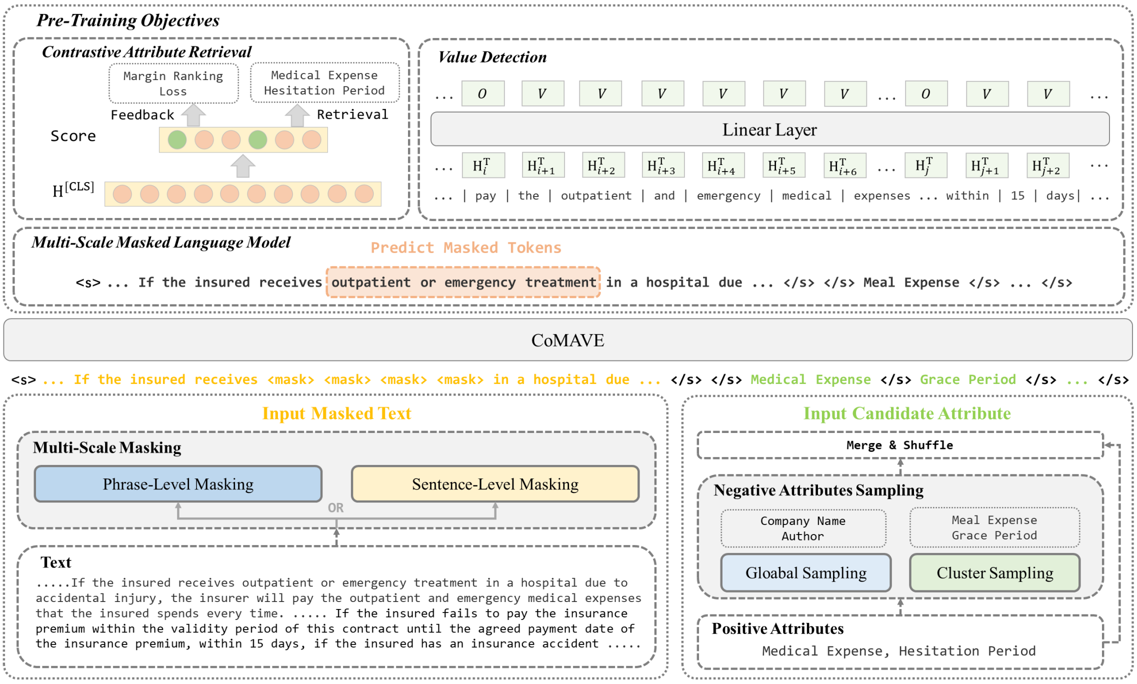

Pre-Training Objectives
Contrastive Attribute Retrieval
Margin Ranking Loss
Medical Expense Hesitation Period
Feedback
Retrieval
Score
H[CLS]
Value Detection
... O V V V V V V ... O V V ...
Linear Layer
... pay | the | outpatient | and | emergency | medical | expenses ... within | 15 | days| ...
Multi-Scale Masked Language Model
Predict Masked Tokens
<s> ... If the insured receives outpatient or emergency treatment in a hospital due ... </s> </s> Meal Expense </s> ... </s>
CoMAVE
<s> ... If the insured receives <mask> <mask> <mask> <mask> in a hospital due ... </s> </s> Medical Expense </s> Grace Period </s> ... </s>
Input Masked Text
Multi-Scale Masking
Phrase-Level Masking
Sentence-Level Masking
OR
Text
.....If the insured receives outpatient or emergency treatment in a hospital due to accidental injury, the insurer will pay the outpatient and emergency medical expenses that the insured spends every time. ..... If the insured fails to pay the insurance premium within the validity period of this contract until the agreed payment date of the insurance premium, within 15 days, if the insured has an insurance accident .....
Input Candidate Attribute
Merge & Shuffle
Negative Attributes Sampling
Company Name Author
Meal Expense Grace Period
Gloabal Sampling
Cluster Sampling
Positive Attributes
Medical Expense, Hesitation Period

Figure 2: An overview of CoMAVE. First, text $\mathcal{T}$ randomly selects one of the masking mechanisms (Phrase-Level Masking or Sentence-Level Masking), meanwhile, based on the golden positive attribute, global sampling and clustering sampling are adopted for negative attributes construction. They are combined and input to CoMAVE for encoding with RoBERTa form. Thereafter, three objectives: **CAR**, **MSMLM**, and **VD** are predicted separately.

each *predicate* and *object* as the attribute a_i and the value v_i, respectively, thereby building a seed set $\{(a_i, V_i)\}_N$ by aligning and merging the attributes. Finally, we use this set as a distant supervision to mine the corresponding texts from the web data, thus building the pre-training corpus.

3.2 Pre-training CoMAVE

Since RoBERTa (Liu et al., 2019) has been shown to be promising and robust on multiple NLP tasks, we use it to initialize CoMAVE. We then further pre-train CoMAVE on our corpus $\mathcal{C}$. As shown in Figure 2, we flatten each pair $(\mathcal{T}, \mathcal{A})$ into a sequence $\mathcal{X}$ with the </s> token,

$$\begin{aligned}&\texttt{<s>}, x_1, x_2, ..., x_n, \texttt{</s>}, \texttt{</s>}, x^a_{1,1}, x^a_{1,2},\\ &x^a_{1,|a_1|}, \texttt{</s>}, x^a_{2,1}, ..., x^a_{m,|a_m|}, \texttt{</s>},\end{aligned} \tag{1}$$

then CoMAVE converts the each token of $\mathcal{X}$ into a semantic vector,

$$\begin{aligned}&\mathbf{h}^{\texttt{<s>}}, \mathbf{h}^{\mathcal{T}}_1, \mathbf{h}^{\mathcal{T}}_2, ..., \mathbf{h}^{\mathcal{T}}_n, \mathbf{h}^{\texttt{</s>}}, \mathbf{h}^{\texttt{</s>}}, \mathbf{h}^a_{1,1}, \mathbf{h}^a_{1,2},\\ &\mathbf{h}^a_{1,|a_1|}, \mathbf{h}^{\texttt{</s>}}, \mathbf{h}^a_{2,1}, ..., \mathbf{h}^a_{m,|a_m|}, \mathbf{h}^{\texttt{</s>}},\end{aligned} \tag{2}$$

where $\mathbf{h}^{\mathcal{T}}_i \in \mathbb{R}^d$ and $\mathbf{h}^a_{j,k} \in \mathbb{R}^d$ denote the vector of x_i and $x^a_{j,k}$, respectively, and $\mathbf{h}^{\texttt{<s>}} \in \mathbb{R}^d$ is regarded as the global semantic vector of $\mathcal{X}$. Considering the above challenges, we design three objectives to pre-train CoMAVE as follows.

Multi-Scale Masked Language Model

The most common objective of pre-training is to employ MLM to guide the model to perform extensively. Unlike BERT or RoBERTa which focuses on token-level recovery, we prefer CoMAVE to be aware of various values, regardless of their scales. Consequently, we design two parallel masking mechanisms, namely phrase-level and sentence-level masking. During pre-training, each $\mathcal{T}$ is performed by only one of the two mechanisms, and the probabilities are set as ρ and 1-ρ, respectively. Furthermore, we empirically find that an appropriate masking percentage is a prerequisite for MLM to be effective. We denote this budget percentage by μ_p and μ_s, respectively, and try to make the masking result close to it for both mechanisms.

In phrase-level masking, we are inspired by SpanBERT (Joshi et al., 2020) and randomly mask a short span of tokens for each selected $\mathcal{T}$ until the budget μ_p is spent. The probability distribution of the masking length, denoted by $l \in [1, l_{\max}]$, is:

$$P_{\text{phrase}}(l) = \frac{\sigma^{-l} + \gamma}{\sum_{l'=1}^{l_{\max}} \sigma^{-l'} + \gamma}, \tag{3}$$

where both σ and γ are hyper-parameters. This distribution ensures that the masking probability of each span decreases smoothly as its length increases, while also preventing long spans from being rarely selected. Note that we make sure that each masked span is formed by complete words.

In sentence-level masking, we mask only one sentence for each selected $\mathcal{T}$ because recovering a sentence requires sufficient context. In this way, it is more difficult to make the total number of masked tokens approach μ_s compared to masked phrases since the length of different sentences can vary significantly. To achieve this goal, we propose a simple but effective strategy to dynamically control the masking probability of each sentence. Specifically, assuming that the current sentence masking rate of $\mu_c = \frac{\sum |\mathcal{T}_{\text{mask}}|}{\sum l(\mathcal{T})}$, where $\mathcal{T}_{\text{mask}}$ is the tokens that has been masked. If $\mu_c < \mu_s$, it means that the current masking rate is less than the standard value, so we should pay more attention to longer sentences $\mathcal{S}_{\text{long}} = \{s | l(s) > l(\mathcal{T}) * \mu_s\}$, giving higher masking probabilities. Otherwise, we should focus on short ones $\mathcal{S}_{\text{short}} = \{s | l(s) \leq l(\mathcal{T}) * \mu_s\}$.

Following BERT (Devlin et al., 2019), we replace 80% of the masked tokens with <mask>, 10% with the random tokens in the corpus, and leave the remaining 10% unchanged.

Contrastive Attribute Retrieval

We expect to adapt CoMAVE to the subtle differences between attributes in the pre-training phase. To this end, for each training text $\mathcal{T}$ and its ground truth attributes $\mathcal{A}^+$, a challenging negative set $\mathcal{A}^- = \mathcal{A}_c^- \cup \mathcal{A}_g^-$ is built to confuse the model. Here, each $a_c \in \mathcal{A}_c^-$ is sampled using clustering to guarantee it is highly similar to $\mathcal{A}^+$ (see below for details), and each $a_g \in \mathcal{A}_g^-$ is a random one from the total attribute pool to maintain the diversity of the negative examples. If $\mathcal{T}$ has no ground truth, then $\mathcal{A}^- = \mathcal{A}_g^-$. During pre-training, CoMAVE is required to retrieve each correct attribute $a^+ \in \mathcal{A}^+$ by scoring all $a \in \mathcal{A}^+ \cup \mathcal{A}^-$ with

$$\mathcal{P}(\mathcal{T}, a) = \text{sigmoid}(\mathbf{h}^{\text{<s>}} * W^{\text{CAR}}), \quad (4)$$

where $W^{\text{CAR}} \in \mathbb{R}^{d*\eta}$ is a trainable parameter and η denotes the maximum of $|\mathcal{A}|$.

To make the score of $\mathcal{A}^+$ higher than that of each negative example, i.e., $\forall a^- \in \mathcal{A}^-$, $\mathcal{P}(\mathcal{T}, a^+) > \mathcal{P}(\mathcal{T}, a^-)$, where $a^+ \in \mathcal{A}^+$. Inspired by (Khosla et al., 2020), we define a Margin Ranking Loss to better leverage contrastive learning and strengthen the distinction between fine-grained attributes,

$$\mathcal{L}_{\text{CRA}} = \sum_{i=1}^{|\mathcal{A}|} \sum_{j=i+1}^{|\mathcal{A}|} (1 - z) * |\mathcal{P}_i - \mathcal{P}_j| + z * \max(0, \lambda - |\mathcal{P}_i - \mathcal{P}_j|), \quad (5)$$

where $\mathcal{P}_i$ is short for $\mathcal{P}(\mathcal{T}, a_i)$, and λ is the margin. If both a_i and a_j are positive or negative examples, $z = 0$, otherwise $z = 1$.

The key to this training objective is how to collect $\mathcal{A}_c^-$ that is highly similar to $\mathcal{A}^+$. Clustering has been proven to have a natural advantage in retrieving similar instances, so we used the widely used *K-medoids* (Park and Jun, 2009) clustering method to construct $\mathcal{A}_c^-$. Concretely, the distance between two attributes is

$$\text{d}(a_i, a_j) = \omega * \tilde{\text{d}}(a_i, a_j) + (1 - \omega) * \tilde{\text{d}}(\mathcal{V}_i, \mathcal{V}_j), \quad (6)$$

$$\tilde{\text{d}}(z_i, z_j) = \tau(f_t(z_i, z_j)) + \tau(f_s(\mathbf{z}_i, \mathbf{z}_j)), \quad (7)$$

where f_t and f_s denote *Levenshtein distance* and *Euclidean metric*, respectively. $\mathbf{z} \in \mathbb{R}^d$ is the RoBERTa (Liu et al., 2019) pooling vector of z. τ denotes the score normalization to ensure balance. The distance considers both the literal and semantic features of the attributes and associated values.

Value Detection

To further cross the gap between the pre-training and downstream AVE tasks, we also add a training objective of detecting values. For $(\mathcal{T}, \mathcal{A})$, wherein each positive attribute $a_i^+ \in A^+$ corresponds to one or more extractable values $\mathcal{V}_i = \{v_1, v_2, ..., v_n\}$ in $\mathcal{T}$. The model needs to classify each token $x_i \in \mathcal{T}$, according to whether it is a part values of $\mathcal{V}$:

$$P(x_i | \mathcal{T}, \mathcal{A}) = \text{softmax}(\mathbf{h}_i^{\mathcal{T}} \cdot W^{\text{VD}}), \quad (8)$$

where $W^{\text{VD}} \in \mathbb{R}^d$ is trainable parameter. We define "V" and "O" as labels to represent that $x_i \in \mathcal{V}$ and $x_i \notin \mathcal{V}$, respectively. Note that each token does not need to be classified to the exactly belonged attribute.

3.3 Fine-tuning

To fully evaluate the effectiveness of our pre-training for downstream tasks, we add the following two output layers to fine-tune our CoMAVE, respectively.

Sequence Tagging Layer

In this setting, $\mathcal{T}$ and all candidate attributes $\mathcal{A}$ are first fed to CoMAVE, as in the pre-training. Then, according to the output $\mathbf{h}^{\mathcal{T}}$, a Conditional Random Field (CRF) generates a sequence $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$. Here n is the length of $\mathcal{T}$ and each $y_i \in \bigcup_{k=1}^{|\mathcal{A}|}\{\mathtt{B}_k, \mathtt{I}_k, \mathtt{O}\}$ is a tag indicating whether the token $x_i \in \mathcal{T}$ is the beginning ($\mathtt{B}_k$), inside ($\mathtt{I}_k$) and outside ($\mathtt{O}_k$) of a value in the attribute $a_k \in \mathcal{A}$.

Machine Reading Comprehension Layer

In this case, CoMAVE takes each $(\mathcal{T}, a_i)$ as input and predicts the span of target values belonging to $a_i \in \mathcal{A}$ in $\mathcal{T}$. Here, we follow a representative work (Li et al., 2020) that consists of two steps. First, the candidate start and end indexes of the span are predicted using the binary classification of each token separately. Subsequently, a matching score is performed for each candidate index pair of start and end. Finally, the pairs with scores above the threshold are retained as the results.

4 Experiments

Datasets

To comprehensively evaluate our method, we used the following four datasets covering both English and Chinese: 1) **INS** is a Chinese AVE dataset which is collected from the real product data of Alipay[2] platform. It contains various types of large-scale insurance products from real scenarios, including wealth insurance, health insurance, travel insurance, life insurance, etc. From each product document, the attributes and values are manually annotated. There are 29 global attributes and the samples are divided into 9112/1138/1138 for Train/Val/Test, respectively. Table 1 gives several groups of similar attributes and the number of their corresponding examples. Table 2 shows the distribution of different value scales. They reveal that the two challenges we focus on are prevalent in INS. 2) **MEPAVE** (Zhu et al., 2020) is a Chinese AVE dataset with examples from the JD e-commerce platform [3], containing 26 global attributes and 87,194 samples. Most of the text is mainly from the product titles. We randomly divided the dataset into three parts of Train/Val/Test in the ratio of 8:1:1 according to (Zhu et al., 2020) for experiments. 3) **AE-Pub** (Xu et al., 2019) is an English AVE dataset with 110,484 samples and over 2400 attributes obtained from AliExpress[4]. In order to make a fair comparison with previous models that could not handle a large number of attributes, we selected 4 frequent attributes (i.e. *BrandName*, *Material*, *Color*, *Category*) and divided the relevant instances randomly by 7:1:2, referring to the dataset publisher. 4) **MAE** (IV et al., 2017) is an English multi-modal AVE dataset that contains 200 million samples and 2000 attributes. Following (Zhu et al., 2020), we built an MAE-text dataset to focus on the textual modal. Same as **AE-Pub**, we also selected the 20 most frequent attributes from Train/Val/Test sets.

FG Attributes Group	Train	Val	Test
Period: hesitation period, grace period, waiting period for continuous insurance, etc.	555	73	77
Age: Maximum insurance age, Minimum insurance age, Maximum renewal age, etc.	544	65	69
Amount: deductible, insured amount, etc.	170	19	24
Area: insured areas, restricted areas, etc.	83	12	13
Disease: disease, disease restriction, etc.	329	39	38

Table 1: Statistical results of the fine-grained attributes in INS. There are about 20% samples containing two or more attributes in the same group.

Length	Train	Val	Test
[1, 5]	5235 (55.0%)	667 (54.8%)	662 (53.5%)
(5, 10]	1622 (17.0%)	202 (16.6%)	207 (16.7%)
(10, 20]	1481 (15.6%)	198 (16.3%)	205 (16.6%)
(20, $+\infty$)	1179 (12.4%)	149 (12.3%)	164 (13.2%)

Table 2: Statistical results of multi-scale value in INS. Note that the results are the amounts of the values

Evaluation Metrics

In most experiments, we used Mirco-F1 scores as the main evaluation metric. We followed the criterion of exact matching, where the complete sequence of predicted attributes and extracted values must be correct. Accuracy was also used as another evaluation in the detailed analysis.

Methods for Comparison

We compared the proposed method with notable AVE methods, including BiLSTM+CRF (Ma and Hovy, 2016), OpenTag (Zheng et al., 2018),

[2] https://www.alipay.com/

[3] https://www.jd.com

[4] https://www.aliexpress.com

Method	INS *Attr*	INS *Val*	INS *Over*	MEPAVE *Attr*	MEPAVE *Val*	MEPAVE *Over*	AE-Pub *Attr*	AE-Pub *Val*	AE-Pub *Over*	MAE *Attr*	MAE *Val*	MAE *Over*
BiLSTM + CRF	74.37	55.26	52.05	89.29	80.49	78.94	81.74	77.03	75.69	79.84	77.40	73.50
OpenTag	73.66	62.65	57.16	88.54	83.26	84.11	86.35	85.18	83.37	82.22	79.34	76.23
ScalingUp	83.88	66.78	65.99	93.42	91.20	89.56	89.05	88.64	87.19	89.36	79.69	78.75
JAVE	87.11	72.40	70.07	95.56	92.98	91.03	90.57	90.14	88.14	93.50	94.12	91.96
AVEQA	86.53	71.34	68.89	95.75	93.65	91.69	91.45	92.86	90.35	94.56	95.78	92.91
UIE	87.46	73.11	71.65	96.67	93.24	92.98	94.35	91.10	89.36	96.02	95.99	94.50
CoMAVE + Tagger	87.31	75.95	73.34	96.02	94.52	93.41	93.07	92.73	90.91	96.31	96.88	94.91
CoMAVE + MRC	**88.90**	**78.70**	**75.92**	**97.04**	**95.78**	**95.39**	**95.97**	**94.24**	**93.65**	**96.92**	**98.12**	**96.55**

Table 3: Overall results compared with existing baselines. Here, *Attr*, *Val*, and *Over* denote the Mirco-F1 of attribute retrieval, value extraction, and overall task, respectively.

SUOpenTag (Xu et al., 2019), JAVE (Zhu et al., 2020), AVEQA (Wang et al., 2020), and UIE (Lu et al., 2022). In addition, the existing representative PLMs are involved in comparison to showing the improvement of our PLM on the AVE task, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2020), MacBERT (Cui et al., 2020), and ELECTRA (Clark et al., 2020).

Implementation Details

Our method ran on Tesla A100 GPUs. All the pre-trained models used in our experiments were large versions by default. Chinese and English versions of CoMAVE were pre-trained respectively for evaluation in two languages. The hyper-parameters in pre-training were set as follows: (1) The batch size and the learning rate were set to 256 and 1e-5. (2) In the CAR task, η, λ, and ω were set to 12, 2, and 0.4, respectively. The ratio for $\mathcal{A}^+$, $\mathcal{A}_g^-$, and $\mathcal{A}_c^-$ was 1:1:1 (3) In the MSMLM task, ρ, σ, γ, ℓ_{max}, μ_p, and μ_s were separately set to 0.2, 1.20, 2e-4, 20, 15%, 10%. In phrase-level masking, ℓ_{max} was set to 20, and ℓ_{mean} was approximately equal to 5.87. In the fine-tuning stage, the batch size and the learning rate were set to 80 and 2e-5, respectively.

4.1 Overall Results

Comparison with AVE Baselines

We first compared with the baselines. To ensure fairness in the number of parameters, we replaced BERT-Base with RoBERTa-Large in the evaluations of Chinese datasets, and the distilled context layer of AVEQA is also replaced by RoBERTa-Large in all evaluations. The results are shown in Table 3. Our proposed CoMAVE equipped with the MRC layer achieves state-of-the-art on all four benchmarks. Most baselines perform poorly on INS because they focus on traditional e-commerce products and cannot handle the two challenges mentioned in Section 1. Unlike them, UIE achieves competitive results, especially on *Attr*, as it is a generic approach pre-trained by multiple information extraction tasks. However, limited by its weak multi-scale value extraction capability, it still cannot handle INS. The performance on *Attr* and *Val* demonstrates that our method brings significant improvements in both attribute retrieval and value extraction, thus outperforming all baselines on *Over*. Benefiting from the pre-training, our CoMAVE outperforms all the baselines by adding only a simple fine-tuning output layer.

Method	INS	MEPAVE	AE-pub	MAE
	+ Tagger Layer			
BERT	70.42*	89.77*	86.72	92.71
RoBERTa	71.63	90.82	88.55	93.12
SpanBERT	-	-	88.23	92.79
MacBERT	71.55	90.79	-	-
ELECTRA	71.69	91.53	88.75	93.42
CoMAVE	**73.34**	**93.41**	**90.91**	**94.91**
	+ MRC Layer			
BERT	71.59*	94.03*	90.49	93.31
RoBERTa	72.89	94.74	91.13	94.14
SpanBERT	-	-	90.62	94.24
MacBERT	73.03	93.99	-	-
ELECTRA	73.46	94.21	91.50	95.32
CoMAVE	**75.92**	**95.39**	**93.65**	**96.55**

Table 4: Overall results compared with existing pre-training model. "*" represents the results run by the base version of the pre-training model.

Comparison with PLMs

To further evaluate the contribution of our pre-training methods, we compared CoMAVE with several common PLMs. All the models were guaranteed to be equipped with the same output layer

Method	INS	MEPAVE	AVE-Pub	MAE
CoMAVE	**75.92**	**95.39**	**93.65**	**96.55**
– MSMLM	74.60	94.87	91.99	94.86
MSMLM – PhraM	75.19	94.97	92.59	95.33
MSMLM – SentM	75.23	95.04	92.77	95.54
– CAR	73.97	94.45	91.32	94.97
CAR – MRL	74.55	95.12	92.58	95.29
CAR – CS	74.30	95.04	91.96	95.12
– VD	74.82	94.74	91.80	95.01
RoBERTa	73.22	94.03	89.49	94.31

Table 5: Overall ablation results on four datasets.

when compared. The results are shown in Table 4. Here SpanBERT and MacBERT have no results on some datasets because of lacking a corresponding language version. SpanBERT achieves almost the same results as RoBERTa with half the number of parameters because it excels in span representation. ELECTRA adopts creative adversarial pre-training and therefore performs well. Compared to the pre-training backbone RoBERTa, our further pre-trained CoMAVE gains a significant improvement. Moreover, regardless of the simple fine-tuning layer used, our model outperforms all the other PLMs, which indicates that our pre-training effectively alleviates the challenges of AVE tasks.

4.2 Ablation Tests

To evaluate the contributions of each training objective, we considered the following settings:

- *– MSMLM*: Removing the training objective of Multi-Scale Masked Language Model.
- *MSMLM – PhraM*: Only Using the sentence-level masking mechanism.
- *MSMLM – SentM*: Only Using the phrase-level masking mechanism.
- *– CAR*: Removing the training objective of Contrastive Attribute Retrieval.
- *CAR – MRL*: Replacing the Margin Ranking Loss $\mathcal{L}_{\text{CAR}}$ with the Cross Entropy Loss.
- *CAR – CS*: Cluster sampling is not used in CAR, i.e., $\mathcal{A}^- = \mathcal{A}_g^-$.
- *– VD*: Removing the training objective of Value Detection.

Here, MRC was uniformly selected as the output layer for all the settings due to its better performance. Table 5 shows the results on four datasets. CAR, MSMLM, and VD all bring obviously improvements, proving the effectiveness and necessity of our pre-training objectives for the AVE tasks.

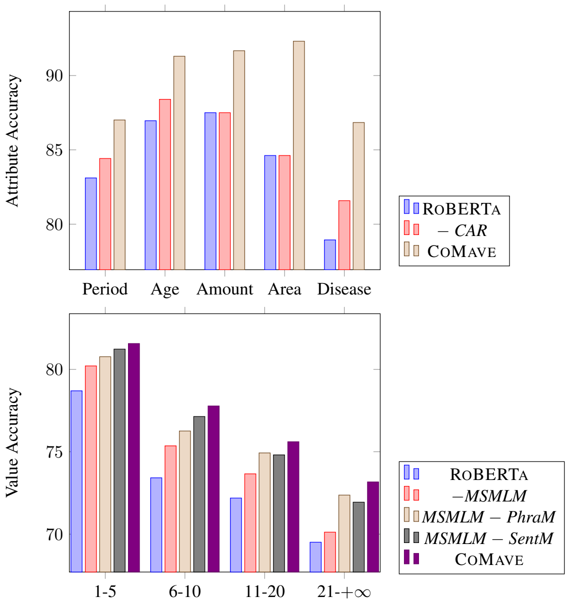


Figure 3: Detailed ablation tests of fine-grained attribute retrieval and multi-scale value extraction on INS. Here, accuracy is used as the evaluation metric.

The result indicates that the contribution of CAR is the most pronounced among the three objectives. The final performance of the model decreases significantly when either the clustering sampling or the contrast loss is removed. In addition, we find that the combination of phrase-level and sentence-level masking is more effective than using only one of them. VD also delivers a promising improvement which proves the benefit for AVE tasks.

4.3 Tests on Fine-Grained Attribute Groups

To further validate the effectiveness of our method in discriminating fine-grained similar attributes, we evaluated the performance of the model on the fine-grained attribute groups mentioned in Table 1. The experimental results are shown in the upper part of Figure 3. Our CoMAVE equipped with all components achieves the best results on all attribute groups. When the CAR training objective is removed, the overall performance shows a dramatic decrease in all the fine-grained attribute groups. This reveals that contrastive learning in a challenging set during pre-training contributed significantly to enhancing the capability of discriminating similar attributes in downstream tasks.

4.4 Performance on Multi-Scale Values

We also tested the performance of the model for extracting values at different scales, and the results are shown in the lower part of Figure 3. As we expected, the contribution of phrase-level masking

Dataset	Setting	RoBERTa	CoMave
INS(29Attr)	5-SHOT	45.47	**50.56**
	10-SHOT	50.91	**54.32**
MEPAVE(26Attr)	5-SHOT	55.53	**58.42**
	10-SHOT	61.86	**64.70**
AE-Pub(4Attr)	5-SHOT	45.84	**51.24**
	10-SHOT	62.31	**66.04**
MAE(20Attr)	5-SHOT	51.07	**55.71**
	10-SHOT	59.19	**62.36**

Table 6: Results of few-shot tests on all datasets.

is greater when dealing with shorter values (number of tokens less than 10). The improvement of sentence-level masking becomes significant when the length of value gradually grows to more than 20. This proves the reasonableness of our combination of phrase-level and sentence-level masking. Moreover, we find that even pre-training without MSMLM, CoMave still performs better than pure RoBERTa. This demonstrates the boost from objective VD and the expected external knowledge of the pre-training corpus.

4.5 Few-Shot Tests

To further fit realistic applications, we also tested the performance of the model in few-shot scenarios. We adopted the N-WAY K-SHOT setup, i.e., the few-shot training set has N attributes, and each attribute has corresponding K training samples randomly selected. Here, we let N equal the number of attributes per dataset and focused on testing two sets of 5-SHOT and 10-SHOT settings.

Table 6 shows the experimental results. Under the stringent condition of using only 5 training samples for each attribute, CoMave scores Mirco-F1 over 50% on all datasets. When the training set is expanded to 10-SHOT, the performance reaches approximately 65% on the other three datasets, except for the challenging INS. The smaller the sample size, the greater the improvement of CoMave. Due to pre-training with a large-scale AVE corpus collected, CoMave is more capable than RoBERTa in handling the few-shot AVE task.

5 Related Work

With the development of e-commerce, Attribute Value Extraction which aims to retrieve the attributes and extract the values from the target data resource in order to obtain the structured information of the products recently attracts lots of attention. Several previous methods (Zheng et al., 2018; Xu et al., 2019) employ traditional sequence tagging models. Furthermore, AVEQA (Wang et al., 2020) first tries to use MRC based method to handle the task, but it can not be applied when each attribute has several different values. JAVE (Zhu et al., 2020) designs a multi-task model which divides the task into two sub-task: attributes prediction and value extraction. AdaTag (Yan et al., 2021) uses a hyper-network to train experts' parameters for each attribute in order to build an adaptive decoder. QUEACO (Zhang et al., 2021) adopts a teacher-student network to leverage weakly labeled behavior data to improve performance. MAVEQA (Yang et al., 2022) mixes multi-source information with a novel global and local attention mechanism. However, none of the existing methods pay attention to the two challenges mentioned in section 1.

Language model pre-training (Devlin et al., 2019; Liu et al., 2019) and task-specific fine-tuning achieve significant improvement on many NLP tasks. Recently, some work (Joshi et al., 2020; Clark et al., 2020; Cui et al., 2020; Sanh et al., 2019) further modified the MLM to achieve better results. In information extraction tasks, UIE (Lu et al., 2022) is proposed as a universal pre-training model for several extraction tasks by generation, it is generic but lacks further fitting for different extraction tasks. Currently, there is no task-specific pre-training model for attribute value extraction.

6 Conclusion

In this paper, we presented a new pre-training model for attribute value extraction, called CoMave which is pre-trained by three novel objectives with a large-scale corpus. Multi-Scale Masked Language Model is designed to force the model to understand multi-scale values by recovering masked spans at both the phrase and sentence levels. Contrastive Attribute Retrieval improves the discrimination of fine-grained attributes based on contrastive learning. Meanwhile, Value Detection is adopted to reinforce the value extraction and further benefit downstream AVE tasks. Extensive experiments indicate that CoMave achieves state-of-the-art results on four benchmarks compared with the existing baselines and PLMs. In future work, we will expand our work on more scenarios and industries, and also explore the optimization of the downstream fine-tune model.

7 Limitations

This paper proposed a novel pre-training model CoMAVE which aims at textual AVE tasks, while in this field, multi-modal AVE tasks also widely exist in many e-commerce platforms. We expect that the following works can leverage CoMAVE as a powerful word embedding pre-training model for text encoding combined with image feature representation in multi-modal AVE tasks in the future. Meanwhile, the same as the previous AVE works, we assume that each $\mathcal{T}$ is an independent extraction object, without considering the context-dependent of the whole data resources, such as long documents and instructions, which exceeds the length of an allowable single input.

Acknowledgements

This work is supported by the NSFC (Grant No. U21A20488, 62006040), the Project for the Doctor of Entrepreneurship and Innovation in Jiangsu Province (Grant No. JSSCBS20210126), the Fundamental Research Funds for the Central Universities, and ZhiShan Young Scholar Program of Southeast University.

References

Min Cao, Sijing Zhou, Honghao Gao, and Youhuizi Li. 2018. A novel hybrid collaborative filtering approach to recommendation using reviews: The product attribute perspective (S). In *The 30th International Conference on Software Engineering and Knowledge Engineering, Hotel Pullman, Redwood City, California, USA, July 1-3, 2018*, pages 7–10. KSI Research Inc. and Knowledge Systems Institute Graduate School.

Huajun Chen, Ning Hu, Guilin Qi, Haofen Wang, Zhen Bi, Jie Li, and Fan Yang. 2021a. Openkg chain: A blockchain infrastructure for open knowledge graphs. *Data Intell.*, 3(2):205–227.

Yongrui Chen, Huiying Li, Guilin Qi, Tianxing Wu, and Tenggou Wang. 2021b. Outlining and filling: Hierarchical query graph generation for answering complex questions over knowledge graph. *CoRR*, abs/2111.00732.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 657–668. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Robert L. Logan IV, Samuel Humeau, and Sameer Singh. 2017. Multimodal attribute extraction. In *6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS 2017, Long Beach, California, USA, December 8, 2017*. OpenReview.net.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland,* *May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Alessandro Magnani, Feng Liu, Min Xie, and Somnath Banerjee. 2019. Neural product retrieval at walmart.com. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 367–372. ACM.

Hae-Sang Park and Chi-Hyuck Jun. 2009. A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.*, 36(2):3336–3341.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian M. Suchanek. 2020. YAGO 4: A reasonable knowledge base. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, pages 583–596. Springer.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5214–5223. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4694–4705. 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.

Pan Yang, Xin Cong, Zhenyu Sun, and Xingwu Liu. 2021. Enhanced language representation with label knowledge for span extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4623–4635. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1321–1331. The Association for Computer Linguistics.

Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, and Qiang Yang. 2021. QUEACO: borrowing treasures from weakly-labeled behavior data for query attribute value extraction. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 4362–4372. ACM.

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 & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1049–1058. ACM.

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 2020, Online, November 16-20, 2020*, pages 2129–2139. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7: Limitations

☑ A2. Did you discuss any potential risks of your work?
Section 7: Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1: Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4: Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4: Experiments, Implementation Details

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4: Experiments, Implementation Details

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4: Experiments

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
The packages used in our code are listed in GitHub.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4: Experiments, Datasets. We build a hand-labeled dataset called INS for evaluation.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
We build the manually labeled dataset INS for evaluation, while there is no human participation in other parts.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
We build the manually labeled dataset INS for evaluation, while there is no human participation in other parts.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
The dataset is collected from the platform of our affiliation, and the source is not discussed in this submission for anonymity.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Our dataset contains the product information of the e-commerce platform, not the information of humans, and humans only participate in labeling the dataset.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Our dataset contains the product information of the e-commerce platform, without the information of humans, and humans only participate in labeling the dataset.