Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach

Qifan Wang*,[1], Li Yang*,[1], Bhargav Kanagal[1], Sumit Sanghai[1], D. Sivakumar[1], Bin Shu[2], Zac Yu[2], Jon Elsas[2]
[1]Google Research, Mountain View, USA
[2]Google, Pittsburgh, USA
{wqfcr,lyliyang,bhargav,sumitsanghai,siva,bins,zacyu,jelsas}@google.com

ABSTRACT

Attribute value extraction refers to the task of identifying values of an attribute of interest from product information. It is an important research topic which has been widely studied in e-Commerce and relation learning. There are two main limitations in existing attribute value extraction methods: scalability and generalizability. Most existing methods treat each attribute independently and build separate models for each of them, which are not suitable for large scale attribute systems in real-world applications. Moreover, very limited research has focused on generalizing extraction to new attributes.

In this work, we propose a novel approach for Attribute Value Extraction via Question Answering (AVEQA) using a multi-task framework. In particular, we build a question answering model which treats each attribute as a question and identifies the answer span corresponding to the attribute value in the product context. A unique BERT contextual encoder is adopted and shared across all attributes to encode both the context and the question, which makes the model scalable. A distilled masked language model with knowledge distillation loss is introduced to improve the model generalization ability. In addition, we employ a no-answer classifier to explicitly handle the cases where there are no values for a given attribute in the product context. The question answering, distilled masked language model and the no answer classification are then combined into a unified multi-task framework. We conduct extensive experiments on a public dataset. The results demonstrate that the proposed approach outperforms several state-of-the-art methods with large margin.

KEYWORDS

attribute value extraction, question answering, generalization

ACM Reference Format:
Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, Jon Elsas. 2020. Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403047

* The two authors contribute equally to this work.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08.
https://doi.org/10.1145/3394486.3403047

Figure 1: Examples of product attributes with their corresponding values. Note that there is no value in the second product for the attribute 'model number'.

1 INTRODUCTION

Product attributes form an essential component of e-commerce platforms today. They are used to power faceted search interfaces, backend retrieval, product ranking and recommendation systems. Further, customers use attributes to compare products and make purchase decisions. However, for most retailers, product attributes are often noisy and incomplete with a lot of missing values. Therefore, it is an important research problem to supplement the product with missing values for attributes of interest, especially with attributes and values that we have never seen before. In this work, we focus on extracting product attribute values from unstructured product information, such as titles and descriptions. The problem of attribute value extraction is illustrated in Figure 1. For example, in the first product, given the title and the attribute 'brand', our goal is to extract the value 'PGM'. In the second product, there is no value in the context for attribute 'model number'. In this case, we need to predict no value.

There has been a lot of interest in this topic, and a plethora of research [2, 3, 8, 35, 53] in this area both in academia and industry. Early works to this problem are rule based approaches [4, 9, 43], which utilize domain-specific regular expressions. These methods

are not able to scale to large set of attributes since they need to develop rules for every possible value corresponding to all attributes. For example, the attribute value 'waterproof' may be specified as 'water-proof', 'water proof' or 'rainproof' or in many other ways. Obtaining the list of such synonym phrases for all the attributes is expensive. In addition, one needs to build regular expressions to identify the absence of a value. For example, 'not available in red' corresponds to 'color' not being 'red'. Several other approaches such as [29, 35] formulate the attribute value extraction as an instance of named entity recognition (NER) problem [30], and build extraction models to identify the entities/values from the input text. With the recent advance in natural language understanding, sequence tagging [50, 54] based approaches have been proposed, which achieve promising results. However, these techniques suffer from two major limitations:

- *Scalability* – they do not scale to millions of attributes that are necessary for real world applications. For instance, the AliExpress taxonomy contains thousands of product categories, and a single category, Sports & Entertainment, has over 8.9k unique attributes [50]. Existing methods treat each attribute independently and build one separate model for each of them, which are not suitable for large scale attribute systems.
- *Generalizability* – they do not work well with new attributes and values. With the rapid expansion of e-commerce, new products with new capabilities are being released constantly which means the model needs to gracefully adapt to new attributes.

In this paper, we formulate the attribute value extraction task as an instance of the question answering (QA) task. In recent years, there have been great advances in question answering and reading comprehension [1] with several new datasets: SQUAD [37] and NaturalQuestions [20]. Specifically, given a context (text sequence) and a question, the question answering task is to identify a best span in the context that corresponds to the answer. To extract an attribute value from a product, we treat the product information as context, and turn the attribute into a question. We employ the contextual encoder from BERT [7] to jointly encode both the question and the context with attention mechanism. The same encoder is used for all attributes, which addresses the scalability problem. To generalize to new attributes, we introduce a distilled masked language model (MLM). Unlike standard MLM, our formulation uses knowledge distillation loss to force the model to continue to remember relevant knowledge from the pretrained BERT model. One key property we require for the model is that it should predict no value when the attribute value is not actually present in the specified context. To account for this, we introduce a no-answer classifier to explicitly model such cases. We then develop a multi-tasking approach to integrate all three tasks, i.e., the QA, the distilled MLM and the no-answer classifier, into a unified learning framework. We conduct an extensive set of experiments on a public dataset, which shows superior performance of the proposed approach over several state-of-the-art methods. The experimental results also demonstrate the robustness of our approach. We summarize the main contributions of this work as follows:

- We present a formulation of attribute value extraction as an instance of question answering, which essentially allows us to infinitely scale the number of attributes.
- We introduce a novel distilled masked language model, which improves the generalization of our approach on completely unseen attributes and values. Moreover, we employ a no-answer classifier to enhance the model ability of predicting no-answers.
- We develop a multi-task approach, which incorporates all three tasks together into a unified learning framework.
- We empirically demonstrate significant improvements over several state-of-the-art baselines on a public benchmark for attribute value extraction.

2 PRELIMINARIES

In this section, we briefly review the BERT model [7], which is closely related to our approach.

2.1 BERT

BERT [7] is a language representation model, which stands for Bidirectional Encoder Representations from Transformers. It is designed to train deep bidirectional representations from unlabeled text through a contextual layer, which is composed of stacked attention layers and feed forward networks as shown in Figure 2.

$$\begin{aligned} H_1 &= FFN(MultiHead(E)) \\ H_k &= FFN(MultiHead(H_{k-1})) \end{aligned} \tag{1}$$

where $E = (e_1, \ldots, e_l)$ are the input embeddings of the sequence. H_k is the output embeddings of the *k-th* layer. The multi-head attention [44] is designed to model the contextual relations among the input sequence. The feed forward network is applied to each position separately and identically, which consists of two linear transformations with a ReLU activation in between.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2}$$

where W_1 and W_2 are two parameter matrices in the feed forward network. b_1 and b_2 are bias terms.

The mask language model (MLM) is introduced in BERT as one of its pretraining tasks. It randomly masks some of the tokens, usually 10% to 15%, from the input sequence, and the objective is to predict the original vocabulary id of the masked word based only on its context. Specifically, the MLM adds a classification layer on top of the BERT output. It first multiplies the output vectors by the embedding matrix and transforms them into the vocabulary dimension. The probability of each word in the vocabulary is then calculated with a softmax function. The MLM objective enables the representation to fuse the left and the right context, which allows BERT to pretrain a deep bidirectional contextual encoder. In this way, the rich language knowledge is effectively captured in the pretrained BERT model.

2.2 Question Answering using BERT

The BERT model is pretrained on the BooksCorpus [55] and English Wikipedia, using next sentence prediction and masked word prediction tasks. The pretrained BERT can be applied to various

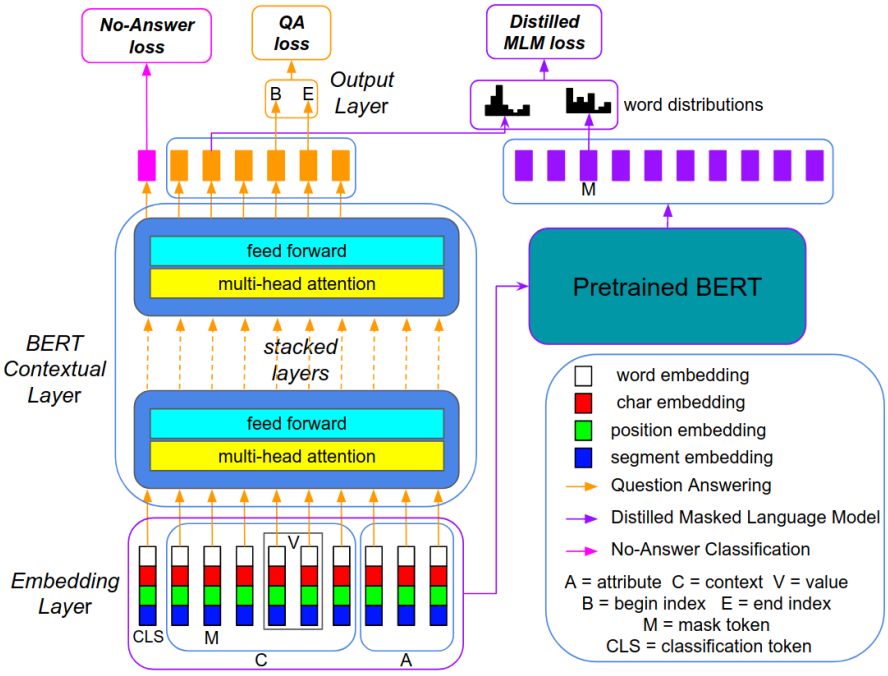

Figure 2: Our AVEQA model architecture.

downstream tasks, such as question answering, document classification and language inference. These tasks use the output of the BERT top layer as the contextual embeddings of the input sequence, and add one additional output layer to conduct the fine-tuning. The pretrained BERT has been successfully applied to the natural question answering task, which achieves state-of-the-art results on the SQuAD benchmark [37]. Due to its superior performance, we adopt the same BERT contextual layer in building our question answering model.

3 AVEQA: ATTRIBUTE VALUE EXTRACTION VIA QUESTION ANSWERING

3.1 Problem Definition

In this section, we formally define the problem of attribute value extraction. Given product context and the attribute, our goal is to extract corresponding attribute value from the context. For instance, the context for the first product in Figure 1 is 'PGM Golf Tower Outdoor Sports Travel Mountaineer Running Comfortable Cotton Golf Towels 5 colors Sports Entertainment for unisex'. We want to extract 'unisex' from the context for attribute 'gender'[1]. Formally, we denote the context as $C = (w_1^c, \ldots, w_n^c)$. Denote the attribute as $A = (w_1^a, \ldots, w_m^a)$. Our model seeks the best value $\bar{V}$ from the context, with its begin and end indices b and e:

$$\begin{aligned} \bar{V} &= \arg\max_{V} \; Pr(\, V \mid C,\, A) \\ &= \arg\max_{b,e} \; Pr(w_b^c, w_e^c \mid C,\, A) \end{aligned} \tag{3}$$

[1]For the sake of simplicity, we focus on single-valued attribute, i.e., there is at most one attribute value in the context. Our work can be easily extend to multi-valued case.

In this work, we formulate the attribute value extraction task as a question answering problem, where each attribute is treated as a question and we seek the best answer span in the context that corresponds to the value.

3.2 Our Multi-task Approach

As aforementioned, existing work has two main limitations - scalability and generalizability. To tackle these two challenges, we propose a novel attribute value extraction approach via question answering using a multi-task framework. The overall model architecture is shown in Figure 2. Essentially, our multi-task model consists of three main components, the question answering (QA), the distilled masked language model (DMLM) and the no-answer (NA) classification. The question answering component aims at finding the best answer span from the context, which answers the question. The distilled masked language model is designed to enhance the model generalization ability, such that it is able to extract values for new attributes. The no-answer classification focuses on identifying those no-answer examples, which further improves the model. The overall objective of our multi-task formulation is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{QA} + \alpha \mathcal{L}_{DMLM} + \beta \mathcal{L}_{NA} \tag{4}$$

where α and β are trade-off parameters to balance the losses among the tasks. In the following sub-sections, we present each component separately in detail.

3.3 Question Answering

The question answering model is composed of three layers: an embedding layer which encodes the input tokens, a contextual layer

that models the complex relationships among the input sequence and an output layer that generates the final results.

3.3.1 Embedding Layer. In the embedding layer, every word in the context and question is converted into a d-dimensional embedding vector. This embedding is obtained by concatenating a word's embedding, a character based embedding, a positional embedding and a segment embedding. Note that we add an additional token, i.e., classification (CLS) token, at the beginning of the input sequence to represent the embedding of the whole sequence. We will provide more details about this special token in the classification model section. The word's embedding and character embedding are well studied in the literature [28]. The positional embedding is employed to inject the order information of the sequence. In this work, we use the absolute position of the words in the sequence [44]. The segment embedding is used to indicate which segment the word is from, i.e., CLS, context or question [7]. In the subsequent sections, we will refer to embedding of a word w_i as e_i=$e(w_i)$. Note that different from previous models, all the embeddings are trainable in our approach. In other words, we only initialize these embeddings from the pretrained BERT model, and allow them to be learned during training instead of fixing them.

3.3.2 Contextual Layer. The contextual layer computes a contextualized representation for every word in the input sequence. In the question answering task, the input sequence to the contextual layer is the concatenation of the context and the question, as well as the CLS token. The output contains a CLS embedding, which represent the whole sequence, and a sequence of contextual embeddings representing the encoded context and question.

The most recent attribute value extraction model [50] employs two separate LSTM-based contextual layers for the context and the question respectively, followed by a cross-attention layer to join the outputs of the two layers. Different from them, we utilize one unique contextual encoder with self-attention mechanism developed in BERT [7]. This contextual encoder structure allows the question and the context to attend each other from the bottom layer to the top layer, and has been widely adopted in recent language models.

3.3.3 Output Layer. The output layer of the question answering model computes the probabilities for the start and end indices of the answer span. A softmax function is applied to the output embeddings to generate the start index. Inspired by the XLNet work [51], we make the end index prediction depend on the start index. Specifically, we concatenate the token embedding of the begin index with every token embedding after it. The new concatenated embedding is then used for finding the best end index. This begin-end dependency modeling is similar to the usage of CRF layer in the open tagging models [50, 54].

$$\begin{aligned} \bar{b} &= \arg\max_i(softmax(W_b H_L^i)) \\ \bar{e} &= \arg\max_{i \geq \bar{b}}(softmax(W_e(Concat(H_L^i, H_L^{\bar{b}})))) \end{aligned} \tag{5}$$

where b and e are the begin and end indices. H_L is the embedding from the contextual layer, and L is the total number of sub-layers. H_L^i is the contextual embedding of the i-th word in the context. W_b and W_e are two output matrices that map the embeddings to the output logits for the begin and end respectively.

3.4 Distilled Masked Language Model

One important factor of a good extraction model is its ability of making accurate prediction on unseen data, which is known as model generalization or zero-shot learning [48]. There are two types of zero-shot problems in attribute value extraction: 1) attributes are not seen in the training set. 2) attribute values are not seen in the training set. The former problem usually implies the latter one. Because if an attribute is not seen during training, its corresponding values are most likely not presented either. Therefore in our following discussion, we focus on addressing the zero-shot problem of new attributes.

Despite the promising results achieved in previous methods, very limited work has been focused on the zero-shot problem of generalizing the model to new attributes, which is one of the main targets in this work. One natural question to ask is: why is the question answering model not able to generalize well? The reason is that the question answering model is designed for large scale attribute value extractions. It is possible for our model to over memorize the training data, especially when training on very large scale data with repeated or similar examples. The learned model is also likely to overfit under large parameter space.

To address this problem, in this work, we introduce a novel distilled masked language model (MLM) to improve the model generalization ability. Instead of predicting the masked word itself, our distilled MLM is aiming at minimizing the cross entropy between the word probability distributions generated from the learned contextual encoder and the pretrained BERT model, using the knowledge distillation loss [23]. Intuitively, the pretrained BERT model contains rich embedding information, which has demonstrated superior performance and been used in various tasks. The distilled MLM ensures our encoder to learn effective contextual representations for new attributes, through masking them out and enforcing the predicted distribution to be consistent with the distribution from the pretrained BERT. In this way, the rich contextual knowledge of the new attributes is transferred from the pretrained BERT to the extraction model, and thus boosts the generalization performance. The knowledge distillation loss is a modified cross entropy loss which is defined as:

$$\mathcal{L}_{DMLM}(Y_{en}, Y_{pre}) = -\sum_{t=1}^{S} \hat{y}_{en}^t \log \hat{y}_{pre}^t \tag{6}$$

$$Y = softmax(W_o(H_M)) \tag{7}$$

where Y_{en} and Y_{pre} are the probability distributions of the masked word generated by the contextual encoder and the pretrained BERT respectively (Eqn.7). S is the vocabulary size. H_M is the output embedding of the masked word. W_o is the output matrix which projects the output embedding to the logits of vocabulary size. $\hat{y}^t$ is the modified probability of the t-th word:

$$\hat{y}^t = \frac{(y^t)^{1/T}}{\sum_j (y^j)^{1/T}} \tag{8}$$

Hinton et al. [10] suggest setting $T > 1$, which increases the weight of smaller logit values and encourages the network to better encode similarities among words. By introducing the distilled MLM loss, our model essentially enforces the two probability distributions of the masked words to be as close as possible, and thus learns better

Attributes	Train	Test
All	88,479	22,005
Brand Name	9,098	2,329
Material	3,100	844
Color	812	184
Category	812	162

Table 1: Statistics of AE-pub with four selected attributes.

methods	$P(\%)$	$R(\%)$	$F_1(\%)$
SUOpenTag [50]	79.85	70.57	74.92
AVEQA	**86.11**	**83.94**	**85.01**

Table 2: Performance comparison of all attributes on AE-pub dataset.

contextual embeddings for the new attributes. We will discuss the impact of the distilled MLM in the experiments.

3.5 No-Answer Classification

One key property we require for the model is that it should predict no value when the attribute value is not actually present in the specified context. To account for this, we employ a no-answer classifier to explicitly model such cases. As shown in Figure 2, we first add a special classification (CLS) token to the input sequence. This CLS token goes through the contextual layer together with all other tokens in the context and question, and attends with them. It can be viewed as a global embedding that represents the whole input sequence. We then apply a binary classifier on top of the contextual embedding of the CLS token to predict whether there is an answer in the context for the question:

$$\bar{y} = max_{y \in \{0,1\}}(softmax(W_{cls} H_L^{cls})) \quad (9)$$

here H_L^{cls} is the output contextual embedding of the CLS token. W_{cls} is the binary classifier.

3.6 Discussion

In this section, we provide discussion that connects our approach with previous methods. Our multi-task formulation is composed of three terms: the QA loss, the distilled MLM loss and the NA loss as presented in Eqn.4. If we remove the distilled MLM and the no-answer classifier by setting both α and β to 0, our model degenerates to the standard question answering model with BERT [7]. If we further replace the BERT contextual layer of the QA component with the BiLSTM layer, our model is regressed to the sequence tagging model in [50]. Moreover, if we also remove the question (attribute) from the QA model, our model is degenerates to the attribute-dependent OpenTag method [54], which is not able to scale to large attribute set. We provide more detailed comparisons with these methods in the experiments section.

4 EXPERIMENTAL RESULTS

4.1 Dataset

We evaluate AVEQA on a public dataset[2], which is collected from AliExpress Sports & Entertainment category [50]. We refer to this dataset as AE-pub in our experiments. The AE-pub dataset contains over 110k examples, i.e., product triples of (context, attribute, value), with more than 2.7k unique attributes and 10k unique values. In addition, there are 21.6k no-answer examples within this dataset, where the value is not present in the context (it is represented as 'NULL' in the dataset). We randomly select 80% of the data, i.e., 88,479 triples, as our training set. The rest 22,005 triples are used for testing. In order to compare with previous sequence tagging models which cannot scale up to huge amounts of attributes, we select a subset of four frequent attributes (i.e., Brand Name, Material, Color and Category) and make comparisons on them. Table 1 shows the statistics and distributions of attributes in AE-pub dataset.

To further examine the generalization ability of our model, we divide the AE-pub dataset into another train and test split by selecting five attributes with relatively low occurrences: Frame Color, Lenses Color, Shell Material, Wheel Material and Product Type. All data triples from these five attributes are put into the test set, while the remaining triples are used for training. In other words, none of these attributes are seen during training. We refer to this data split as AE-zero-shot, as it is designed for evaluating zero-shot extraction.

4.2 Implementation Details

Our models are implemented with Tensorflow and Keras, and each one is trained on TPUs in pod configuration. We initialize our model with 768-dimension pretrained public BERT. The number of layers for the contextual encoder is set to 12. For the multi-head attention layer, the number of heads is set to 12, with 128 maximum sequence length. The number of hidden units in the FFN is set to 3072. The hyper-parameter T in the distilled MLM is set to 2.0.

During training, we use the gradient descent algorithm with Adam [17] optimizer. The initial learning rate is set to $1e^{-5}$. The dropout probability for the attention layer is set to 0.1. The hyper-parameters α is set to 0.5, with β also set to 0.5. We use two different batch sizes, 2048 and 32, on AE-pub and AE-zero-shot datasets respectively. The total number of training steps is set to 200k for all our experiments.

4.3 Evaluation Metrics

We use precision, recall and F_1 score as evaluation metrics denoted as P, R and F_1. We follow Exact Match [37] criteria to compute the scores. We repeat each experiment 10 times and report the metrics based on the average over these runs.

4.4 Baselines

We compare our models with four state-of-the-art baselines on attribute value extraction, BiLSTM [11], BiLSTM-CRF [13], OpenTag [54] and SUOpenTag [50].

- **BiLSTM** [11] uses the word embedding from pretrained BERT to represent each word in the context, then applies BiLSTM to produce the contextual embedding.

[2] https://raw.githubusercontent.com/lanmanok/ACL19_Scaling_Up_Open_Tagging/master/publish_data.txt

	Brand Name			Material			Color			Category		
methods	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
BiLSTM [11]	90.21	90.67	90.44	72.12	62.56	67.00	52.13	48.65	50.33	60.84	50.02	54.89
BiLSTM-CRF [13]	90.45	90.97	90.71	72.40	63.45	67.63	52.68	48.12	50.30	60.48	50.65	55.13
OpenTag [54]	90.32	91.10	90.71	72.56	64.78	68.45	52.83	48.45	50.54	62.17	50.79	55.91
SUOpenTag [50]	91.19	91.57	91.38	74.07	63.86	68.59	57.58	48.72	52.78	62.03	51.58	56.32
AVEQA	**96.41**	**97.00**	**96.70**	**86.34**	**87.20**	**86.76**	**76.47**	**77.68**	**77.06**	**84.43**	**85.70**	**85.05**

Table 3: Performance comparison of four selected attributes on AE-pub dataset.

Attributes	Models	$P(\%)$	$R(\%)$	$F_1(\%)$
Frame Color	SUOpenTag	63.16	48.00	54.55
	AVEQA	**86.54**	**48.82**	**62.20**
Lenses Color	SUOpenTag	64.29	40.91	50.00
	AVEQA	**88.42**	**45.91**	**59.94**
Shell Material	SUOpenTag	54.05	44.44	48.78
	AVEQA	**73.96**	**65.76**	**69.52**
Wheel Material	SUOpenTag	70.59	37.50	48.98
	AVEQA	**70.69**	**65.56**	**67.96**
Product Type	SUOpenTag	64.86	43.29	51.92
	AVEQA	**91.79**	**70.69**	**79.82**

Table 4: Zero-shot extraction results: performance comparison on five new attributes.

- **BiLSTM-CRF** [13] uses a CRF layer on top of the BiLSTM layer to model the association of predicted tags, which is considered to be the pioneer and the state-of-the-art sequence tagging model for NER.
- **OpenTag** [54] adds a self-attention mechanism between the BiLSTM layer and the CRF layer.
- **SUOpenTag** (Scaling Up Open Tag) [50] uses one BiLSTM to produce the contextual word embedding for the context, and another BiLSTM to produce a single embedding for the attribute. A cross attention layer is applied between the context word embedding and the attribute embedding to join the outputs, followed by a CRF layer.

BiLSTM, BiLSTM-CRF and OpenTag are all attribute-dependent methods, which build one separate model for each attribute and thus are not able to scale up. SUOpenTag is designed to extend the OpenTag to deal with large set of attributes.

4.5 Results and Discussion

We conduct four sets of experiments on both AE-pub and AE-zero-shot to evaluate the performance of the proposed AVEQA.

4.5.1 Performance Comparison. We first compare our model with four state-of-the-art attribute value extraction methods as mentioned in Section 4.2. Note that the BiLSTM, BiLSTM-CRF and OpenTag methods are not able to scale up to all attributes on AE-pub. Therefore, we only conduct comparison with them on the four frequent attributes. The evaluation results on AE-pub are reported in Table 2 and 3. From these comparison results, we can see that AVEQA outperforms the other compared methods with large margins. For example, the F_1 metric of AVEQA increases by 13.5% compared with SUOpenTag over all attributes. It increases by 26.7% compared with OpenTag on the Material attribute. There are two main reasons: first, our model employs the attention mechanism with stacked layers in contextual encoder, which allows the context and the question to attend each other from the bottom layer to the top layer, resulting in better contextual embeddings; second, our model allows all the embeddings, i.e., the character embedding, word embedding, position embedding and segment embedding, to be learned during training, while the baseline methods fix all the embeddings after initialization. In this way, our model is able to learn more effective embeddings that are more suitable for the extraction task.

We further conduct zero-shot extraction experiment to evaluate the generalization ability of our model. We compare with SUOpenTag method on the AE-zero-shot dataset, as SUOpenTag is the only baseline that can work on new attributes. The zero-shot extraction results are reported in Table 4. It can be seen that our model achieves much better results compared to SUOpenTag on these five new attributes. The reason is that the distilled MLM effectively transfers knowledge about the new attributes from the pretrained model, which benefits the QA model on zero-shot extractions. Moreover, the no-answer classifier is also effective in predicting those no-answer examples which further boost the performance. We provide more details later on the effect of both distilled MLM and no-answer classifier.

4.5.2 Impact of Multi-task Learning. To evaluate the effectiveness of different components in the multi-task approach, we conduct a set of experiments by removing each component individually from our model. In particular, there are three components, question answering, distilled MLM and no-answer classifier, in our formulation Eqn.4. We train a model by setting both α and β to 0, which is equivalent to removing both the distilled MLM and no-answer classification from the model, and only keep the original question answering part. We name this model QA for later reference. Similarly, by setting only one of α and β to 0, we train another two models, namely QA+CLS and QA+MLM.

The F_1 results of QA, QA+MLM, QA+CLS and AVEQA on both AE-pub and AE-zero-shot are shown in Figure 3. It can be observed from the figure that adding the distilled MLM boosts the performance significantly (e.g., by 5.9%) compared to the basic QA model on zero-shot extractions, while its F_1 score decreases a little bit from 85.08 to 84.97 on AE-pub dataset. This behavior is consistent with our expectation, as the distilled MLM is specifically designed for better model generalization but not for improving the model

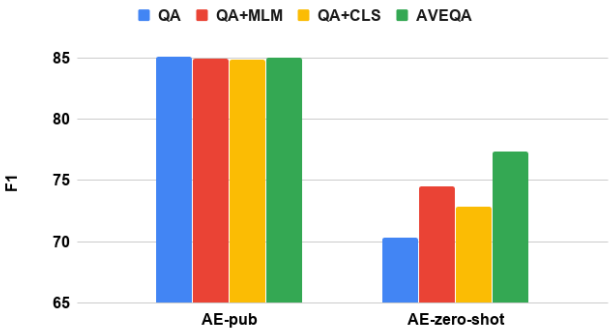

Figure 3: F_1 (%) results of QA, QA+MLM, QA+CLS and AVEQA on both AE-pub and AE-zero-shot.

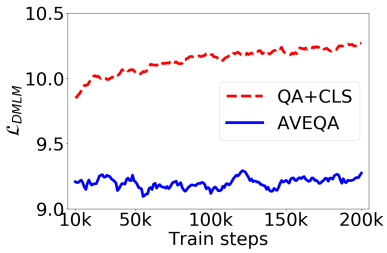

Figure 4: Distilled MLM losses of QA+CLS and AVEQA on AE-zero-shot.

on existing attributes. However, the QA+MLM model is still able to generate comparable results with the QA model on AE-pub. Another observation is that the no-answer classifier also benefits the model, especially on AE-zero-shot data. Our hypothesis is that no-answer classifier can work effectively on those no-answer examples. Finally, it is clear from these experimental results that the AVEQA model, which incorporates all three components, achieves the best results on zero-shot learning.

To further examine the behavior of the distilled MLM, we plot the distilled MLM loss of QA+CLS and AVEQA on AE-zero-shot in Figure 4. It is clear that the distilled MLM loss for QA+CLS continuously goes up during training, while the loss for AVEQA is much lower and relatively stable. This phenomenon illustrates that the AVEQA model preserves the embedding knowledge in the pretrained BERT, and thus is able to generalize well on new attributes. In contrast, without the distilled MLM component, the QA+CLS model gradually forgets the knowledge in the pretrained BERT, and overfits to the training data.

4.5.3 Impact of Training Batch Size and Learning Rate. In this section, we conduct experiments to evaluate the model performance with different training batch size and learning rate. We first vary the training batch size from {32, 256, 2048} by fixing the learning rate to $1e^{-6}$. The F_1 scores with different batch sizes on both datasets are reported in Figure 5. It can be seen from the figure that the batch size does not affect the model performance on AE-pub. However, the model converges faster with larger batch size. We also observe that

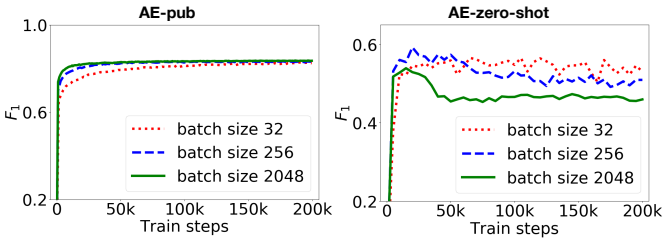

Figure 5: Impact of different training batch size on both AE-pub and AE-zero-shot.

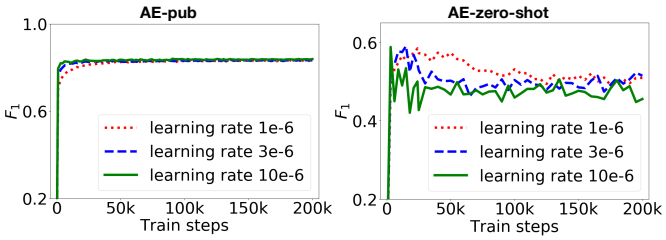

Figure 6: Impact of different learning rate on both AE-pub and AE-zero-shot.

the model performance decreases with large batch size on AE-zero-shot. This observation is consistent with previous work on model generalization [12, 16], which conclude that large batch training increases the generalization gap. Another interesting observation is that the zero-shot model performance with 2048 training batch drops after 20k training steps. Our explanation is that our model is initialized with pretrained BERT, which is able to generalize to new attributes. As the large batch training goes on, our model overfits to the training set which leads to the performance decay.

To explore the effect of different learning rates, we vary the learning rate from $\{1e^{-6}, 3e^{-6}, 10e^{-6}\}$ by fixing the training batch size to 256. We report the F_1 scores with different learning rates on both datasets in Figure 6. It is clear from the figure that the performance of our model is relatively stable with different learning rates on AE-pub, and larger learning rate leads to faster model convergence. On the zero-shot extraction, we observe that the F_1 score of the model with larger learning rate decays faster than the smaller ones.

4.5.4 Parameter Sensitivity. To evaluate the robustness of the proposed approach, we conduct parameter sensitivity experiments with respect to α and β on AE-zero-shot (α and β are not sensitive on AE-pub as shown in Figure 3). In each experiment, we tune only one parameter from {0, 0.1, 0.5, 1, 5, 10}, while fixing the other parameter to the value as described in our implementation details. We report the F_1 results in Figure 7. It is clear from these experimental results that the performance of AVEQA is relatively stable with respect to α and β. From these experiments we found that our model achieves the best scores when $\alpha = 0.5$ and $\beta = 0.5$. We also observe similar results of the proposed method in terms of precision and recall. But due to the limit of space, they are not presented here.

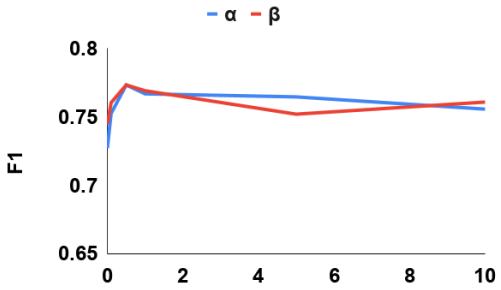

Figure 7: Parameter sensitivity for α and β on AE-zero-shot.

5 RELATED WORK

This section reviews the related work in three research areas: attribute value extraction, question answering and relation learning.

5.1 Attribute Value Extraction

Early works on attribute value extraction use rule-based extraction techniques [9, 30, 43] which use domain-specific seed dictionary or vocabulary to identify key phrases and attributes. Ghani et al. [8] predefine a set of product attributes to extract the corresponding attributes values. Wong et al. [46] extract attribute-value pairs from the semi-structured text, such as tables and list. Shinzato and Sekine [40] propose an unsupervised method to extract attribute values from product description, which filters out sentences with problematic annotations based on statistical measures and morpheme patterns. Several rule-based and linguistic approaches [4, 27] leverage syntactic structure of sentences to extract dependency relations, which do not work well on irregular structures like titles. An NER system was proposed in [35] for extracting product attributes and values. In this work, supervised NER and bootstrapping technology are combined to expand the seed dictionary of attribute values. A similar NER method was built [29] to tag brands in product titles leveraging existing brand values. However, these rule-base and domain-specific methods suffer from limited coverage and closed world assumptions.

With the development of deep neural network, various neural network methods have been proposed and applied in sequence tagging successfully. Ling and Weld [24] apply a multi-label multi-class Perceptron classifier for NER. A linear chain CRF is used to segment text with BIO tagging. Collobert et al. [6] combine deep FFNN and word embedding [28] to explore many NLP tasks including POS tagging, chunking and NER. Huang et al. [13] is the first to apply BiLSTM-CRF model to sequence tagging task. But it employs heavy feature engineering to extract character-level features. Lample et al. [21] utilize BiLSTM to model both word-level and character-level information rather than hand-crafted features, thus construct end-to-end BiLSTM-CRF model for sequence tagging task. Chiu and Nichols [5] model character level information using convolutional neural network (CNN), which achieves competitive performance for two sequence tagging tasks at that time. Ma and Hovy [26] propose an end to end LSTM-CNNs-CRF model.

Recently, several approaches employ sequence tagging model for attribute value extraction. Kozareva et al. [19] adopt BiLSTM-CRF model to tag several product attributes from search queries with hand-crafted features. Furthermore, Zheng et al. [54] develop an end-to-end tagging model utilizing BiLSTM, CRF, and attention mechanism without any dictionary. Most recently, Xu et al. [50] adopt only one global set of BIO tags for any attributes to scale up the model, which explicitly models the semantic representations for attribute and product title. However, most of these methods treat each attribute independently and build one separate model for each of them, which are not suitable for large scale attribute systems. Moreover, model generalization is not considered, which is important in zero-shot extraction.

5.2 Question Answering

Our work is inspired by the recent advance in question answering (QA) [14, 18]. Question answering has been a challenging task in reading comprehension [1] and natural language understanding. We focus on the QA problem of selecting the span of text from a given context that answers a question. The initial question answering systems are based on heuristics [39] and statistical approaches [15], which are domain-specific and not able to generalize to new questions and domains. With the recent development of deep learning, neural network based methods [31, 42, 49] establish a new paradise for question answering. Rajpurkar et al. [37] create a standard question answering dataset, SQuAD, for evaluating different QA methods. Most recently, Devlin et al. [7] propose a pre-training BERT approach using bi-directional encoder representations from transformers, which achieves state-of-the-art results in natural question answering task. A complete review of question answering research can be found in [14, 18].

5.3 Relation Learning

Relation learning (or extraction) [33, 38, 41, 45, 52] refers to the task of extracting relational tuples and putting them in a knowledge base. The tuple usually corresponds to a subject (usually an entity such as a person), a predicate (the relation itself such as 'place of birth') and an object (usually another entity such as the location where the person is born). Attribute value extraction can be thought of as the problem where the subject is known (the product), and given the attribute (i.e., the relation) extract the value. However, relation extraction has traditionally focused on extracting relations from sentences relying on entity linking systems to identify the subject/object and building models to learn the predicates in a sentence [3, 22]. Whereas in attribute value extraction [34, 36], usually the predicates (i.e, the attribute names) rarely occur in the product title or the description, and entity linking is very hard because the domain of all entities/values is unknown. Recently, Lockard et al. [25] propose to generate training labels by aligning an existing knowledge base with a semi-structured web page. A classifier is trained based on the labels to predict new relation instances. Wu et al. [47] design a machine-learning-based knowledge base construction system to extract relations conveyed jointly via textual, structural, tabular, and visual expressions. For a comprehensive review on relation extraction, please refer to [32].

6 CONCLUSION

In this work, we present a novel approach for attribute value extraction via question answering in a multi-task learning framework. We build a question answering model which treats each attribute as a question and finds the best answer span corresponding to the attribute value in the product context. A unique BERT contextual encoder is adopted and shared across all attributes to encode both the context and the question, which makes the model scalable. A distilled masked language model is introduced to improve the model generalization ability. In addition, we employ a no-answer classifier to explicitly handle the missing value cases. The three components are then integrated into a unified multi-task framework. An extensive set of experiments has been conducted on a public dataset. The experimental results demonstrate both the effectiveness and the robustness of the proposed approach, which outperforms several state-of-the-art methods with large margin. There are several possibilities to explore in the future research. For example, we plan to extend our approach to model long text sequence or the whole web page. We also plan to use active learning to improve the zero-shot performance even more.

REFERENCES

[1] R. Baradaran, R. Ghiasi, and H. Amirkhani. A survey on machine reading comprehension systems. *CoRR*, abs/2001.01582, 2020.

[2] D. Carmel, L. Lewin-Eytan, and Y. Maarek. Product question answering using customer generated content - research challenges. In *SIGIR*, pages 1349–1350, 2018.

[3] K. Chen, L. Feng, Q. Chen, G. Chen, and L. Shou. EXACT: attributed entity extraction by annotating texts. In *SIGIR*, pages 1349–1352, 2019.

[4] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *EMNLP*, pages 1002–1012, 2010.

[5] J. P. C. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370, 2016.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[8] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. E. Fano. Text mining for product attribute extraction. *SIGKDD Explorations*, 8(1):41–48, 2006.

[9] V. Gopalakrishnan, S. P. Iyengar, A. Madaan, R. Rastogi, and S. H. Sengamedu. Matching product titles using web-based enrichment. In *CIKM*, pages 605–614, 2012.

[10] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *NIPS*, pages 1731–1741, 2017.

[13] Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[14] K. S. D. Ishwari, A. K. R. R. Aneeze, S. Sudheesan, H. J. D. A. Karunaratne, A. Nugaliyadde, and Y. Mallawarachchi. Advances in natural language question answering: A review. *CoRR*, abs/1904.05276, 2019.

[15] A. Ittycheriah, M. Franz, W. Zhu, A. Ratnaparkhi, and R. J. Mammone. Ibm's statistical question answering system. In *TREC*, 2000.

[16] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[18] L. Kodra and E. K. Meçe. Question answering systems: A review on present developments, challenges and trends. *International Journal of Advanced Computer Science and Applications*, 8(10.14569), 2017.

[19] Z. Kozareva, Q. Li, K. Zhai, and W. Guo. Recognizing salient entities in shopping queries. In *ACL*, 2016.

[20] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *TACL*, 7:452–466, 2019.

[21] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. In *NAACL-HLT*, pages 260–270, 2016.

[22] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *CoNLL*, pages 333–342, 2017.

[23] Z. Li and D. Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2018.

[24] X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*, 2012.

[25] C. Lockard, X. L. Dong, P. Shiralkar, and A. Einolghozati. CERES: distantly supervised relation extraction from the semi-structured web. *PVLDB*, 2018.

[26] X. Ma and E. H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, 2016.

[27] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *EACL*, pages 1–8, 1999.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[29] A. More. Attribute extraction from product titles in ecommerce. *CoRR*, abs/1608.04670, 2016.

[30] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[31] A. Nugaliyadde, K. W. Wong, F. Sohel, and H. Xie. Reinforced memory network for question answering. In *ICONIP*, 2017.

[32] S. Pawar, G. K. Palshikar, and P. Bhattacharyya. Relation extraction : A survey. *CoRR*, abs/1712.05191, 2017.

[33] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W. Yih. Cross-sentence n-ary relation extraction with graph lstms. *TACL*, 5:101–115, 2017.

[34] P. Petrovski and C. Bizer. Extracting attribute-value pairs from product specifications on the web. In *ICWI*, pages 558–565, 2017.

[35] D. Putthividhya and J. Hu. Bootstrapped named entity recognition for product attribute extraction. In *EMNLP*, pages 1557–1567, 2011.

[36] D. Qiu, L. Barbosa, X. L. Dong, Y. Shen, and D. Srivastava. DEXTER: large-scale discovery and extraction of product specifications on the web. *PVLDB*, 8(13):2194–2205, 2015.

[37] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016.

[38] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *ECML/PKDD*, pages 148–163, 2010.

[39] E. Riloff and M. Thelen. A rule-based question answering system for reading comprehension tests. In *ANLP/NAACL Workshop*, pages 13–19, 2000.

[40] K. Shinzato and S. Sekine. Unsupervised extraction of attributes and their values from product description. In *IJCNLP*, pages 1339–1347, 2013.

[41] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *SIGKDD*, 2006.

[42] M. Tan, B. Xiang, and B. Zhou. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108, 2015.

[43] D. Vandic, J. van Dam, and F. Frasincar. Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437, 2012.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[45] Q. Wang, B. Kanagal, V. Garg, and D. Sivakumar. Constructing a comprehensive events database from the web. In *CIKM*, pages 229–238, 2019.

[46] Y. W. Wong, D. Widdows, T. Lokovic, and K. Nigam. Scalable attribute-value extraction from semi-structured text. In *ICDM Workshops*, pages 302–307, 2009.

[47] S. Wu, L. Hsiao, X. Cheng, B. Hancock, T. Rekatsinas, P. Levis, and C. Ré. Fonduer: Knowledge base construction from richly formatted data. In *SIGMOD*, pages 1301–1316, 2018.

[48] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2019.

[49] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, pages 2397–2406, 2016.

[50] H. Xu, W. Wang, X. Mao, X. Jiang, and M. Lan. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *ACL*, pages 5214–5223, 2019.

[51] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*, pages 5754–5764, 2019.

[52] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344, 2014.

[53] J. Zhao, Z. Guan, and H. Sun. Riker: Mining rich keyword representations for interpretable product question answering. In *SIGKDD*, pages 1389–1398, 2019.

[54] G. Zheng, S. Mukherjee, X. L. Dong, and F. Li. Opentag: Open attribute value extraction from product profiles. In *SIGKDD*, pages 1049–1058, 2018.

[55] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27, 2015.