Scaling Up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title

Huimin Xu[1,2], **Wenting Wang**[2], **Xin Mao**[1,2], **Xinyu Jiang**[1], **Man Lan**[1]*
[1] School of Computer Science and Software Engineering, East China Normal University
[2] Alibaba Group
{hmxu,xinmao,xyjiang}@stu.ecnu.edu.cn, mlan@cs.ecnu.edu.cn
{xinjiu.xhm,nantiao.wwt,sunian.mx}@alibaba-inc.com

Abstract

Supplementing product information by extracting attribute values from title is a crucial task in e-Commerce domain. Previous studies treat each attribute only as an entity type and build one set of NER tags (e.g., *BIO*) for each of them, leading to a scalability issue which unfits to the large sized attribute system in real world e-Commerce. In this work, we propose a novel approach to support value extraction scaling up to thousands of attributes without losing performance: (1) We propose to regard attribute as a query and adopt only one global set of *BIO* tags for any attributes to reduce the burden of attribute tag or model explosion; (2) We explicitly model the semantic representations for attribute and title, and develop an attention mechanism to capture the interactive semantic relations in-between to enforce our framework to be attribute comprehensive. We conduct extensive experiments in real-life datasets. The results show that our model not only outperforms existing state-of-the-art NER tagging models, but also is robust and generates promising results for up to $8,906$ attributes.

1 Introduction

Product attributes are vital to e-Commerce as platforms need attribute details to make recommendations and customers need attribute information to compare products and make purchase decisions. However, attribute information is often noisy and incomplete because of the inevitable hurdles posed to retailers by the extremely huge and complex e-Commerce attribute system. On the other hand, product titles which are carefully designed by retailers are packed tightly with details to highlight all important aspects of products. Figure 1 shows the product page of a '*dress*' from AliExpress[1] which is an emerging and fast-growth global e-Commerce platform. The product title "*2019 Summer Women Button Decorated Print Dress Off-shoulder Party Beach Sundress Boho Spaghetti Long Dresses Plus Size FICUSRONG*" contains attribute values: (1) already listed in Item Specifics, such as 'Women' for *Gender*, 'Summer' for *Season*, etc; (2) missing in Item Specifics, such as '2019' for *Year*, 'Plus Size' for *Size*, etc. In this paper, we are interested in supplementing attribute information from product titles, especially for the real world e-Commerce attribute system with thousands of attributes built-in and new attributes and values popping out everyday.

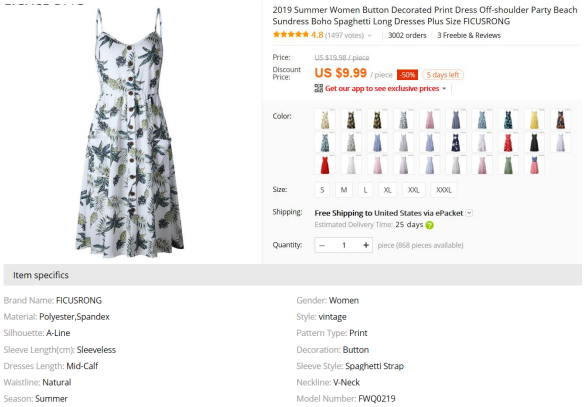

Figure 1: Snapshot of a product page.

Previous work (Ghani et al., 2006; Ling and Weld, 2012; Sheth et al., 2017) on attribute value extraction suffered from Closed World Assumption which heavily depends on certain pre-defined attribute value vocabularies. These methods were unable to distinguish polysemy values such as 'camel' which could be the *Color* for a sweater rather than its *Brand Name*, or find new attribute values which have not been seen before. More recently, many research works (More, 2016; Zheng et al., 2018) formulate attribute value extraction

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

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5214–5223
Florence, Italy, July 28 - August 2, 2019.

problem as a special case of Named Entity Recognition (NER) task (Bikel et al., 1999; Collobert et al., 2011). They adopted sequence tagging models in NER as an attempt to address the Open World Assumption purely from the attribute value point of view. However, such tagging approach still failed to resolve two fundamental challenges in real world e-Commerce domain:

Challenge 1. Need to scale up to fit the large sized attribute system in the real world. Product attribute system in e-Commerce is huge and may overlap cross domains because each industry designs its own standards. The attribute size typically falls into the range from tens of thousands to millions, conservatively. For example, `Sports & Entertainment` category from AliExpress alone contains $344,373$ products (may vary daily) with $77,699$ attributes and $482,780$ values. Previous NER tagging models have to introduce one set of entity tags (e.g., *BIO* tags) for each attribute. Thus, the large attribute size in reality renders previous works an infeasible choice to model attribute extraction. Moreover, the distribution of attributes is severely skewed. For example, 85% of attributes appear in less than 100 `Sports & Entertainment` products. Model performance could be significantly degraded for such rarely occurring attributes (e.g., *Sleeve Style*, *Astronomy*, etc.) due to insufficient data.

Challenge 2. Need to extend Open World Assumption to include new attribute. With the rapid development of e-Commerce, both new attributes and values for newly launched products are emerging everyday. For example, with the recent announcement of '*foldable mobile phone*, a new attribute *Fold Type* is created to describe how the mobile phone can be folded with corresponding new attribute values '*inward fold*', '*outward fold*', etc. Previous NER tagging models view each attribute as a separate entity type and neglect the hidden semantic connections between attributes. Thus, they all fail to identify new attributes with zero manual annotations.

In this paper, to address the above two issues, we propose a novel attribute-comprehension based approach. Inspired by Machine Reading Comprehension (MRC), we regard the product title and product attribute as 'context' and 'query' respectively, then the 'answer' extracted from 'context' equals to the attribute value wanted. Specifically, we model the contexts of title and attribute respectively, capture the semantic interaction between them by attention mechanism, and then use Conditional Random Fields (CRF) (Lafferty et al., 2001) as output layer to identify the corresponding attribute value. The main contributions of our work are summarized as follows:

- **Model.** To our knowledge, this is the first framework to treat attribute beyond NER type alone but leverage its contextual representation and interaction with title to extract corresponding attribute value.

- **Learning.** Instead of the common *BIO* setting where each attribute has its own *BIO* tags, we adopt a novel *BIO* schema with only one output tag set for all attributes. This is enabled by our model designed to embed attribute contextually rather than attribute tag along. Then learning to extract thousands of attributes first becomes feasible.

- **Experiments.** Extensive experiments in real world dataset are conducted to demonstrate the efficacy of our model. The proposed attribute-comprehension based model outperforms state-of-the-art models by average 3% in F_1 score. Moreover, the proposed model scales up to $8,906$ attributes with an overall F_1 score of 79.12%. This proves its ability to produce stable and promising results for not only low and rare frequency attributes, but also new attributes with zero extra annotations.

To the best of our knowledge, this is the first framework to address the two fundamental real world issues for open attribute value extraction: scalability and new-attribute. Our proposed model does not make any assumptions on attribute size, attribute frequencies or the amount of additional annotations needed for new attributes.

The rest of the paper is organized as follows. Section 2 gives a formal problem statement for this task. Section 3 depicts our proposed model in details. Section 4 lists the experimental settings of this work. Section 5 reports the experimental results and analysis. Section 6 summarizes the related work, followed by a conclusion in Section 7.

2 Problem Statement

In this section, we formally define the attribute value extraction task. Given product title T and

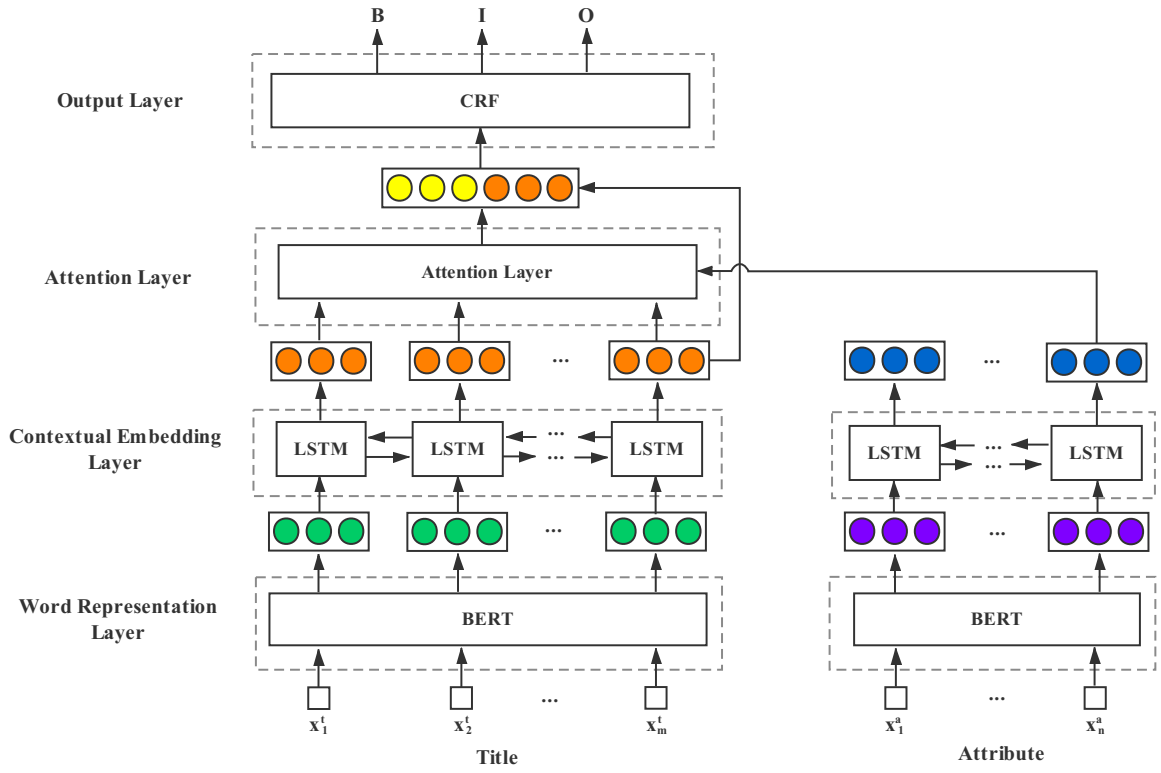

Figure 2: Architecture of the proposed attribute-comprehension open tagging model.

attribute A, our goal is to extract corresponding attribute value for A from T. For example, the title and attributes from Figure 1 are given as below:

- **Product Title:** 2019 Summer Women Button Decorated Print Dress Off-shoulder Party Beach Sundress Boho Spaghetti Long Dresses Plus Size FICUSRONG.
- **Attributes:** Season, Gender, Neckline

Considering the three attributes of interest, i.e., *Season*, *Gender* and *Neckline*, we aim to obtain 'Summer' for *Season*, 'Women' for *Gender* and 'NULL' for *Neckline*, where the former two attributes are described in title but the latter is not presented in title.

Formally, given the product title $T = \{x_1^t, x_2^t, \ldots, x_m^t\}$ of length m and attribute $A = \{x_1^a, x_2^a, \ldots, x_n^a\}$ of length n, our model outputs the tag sequence $y = \{y_1, y_2, \ldots, y_m\}$, $y_i \in \{B, I, O\}$, where B and I denote the beginning and inside tokens for the extracted attribute value respectively, and O denotes outside of the value.

3 Attribute-Comprehension Open Tagging Model

Previous work on sequence tagging built one model for every attribute with a corresponding set of attribute-specific tags. Such approach is unrealistic on real-life large sized attribute set because of two reasons: (1) it is computationally inefficient to model thousands of attributes; (2) very limited data samples are presented for most attributes resulting in non-guaranteed performance. To tackle the two challenges raised in Section 1, we propose a novel attribute-comprehension based open tagging approach to attribute value extraction. Figure 2 shows the architecture of our proposed model. At first glance, our model, adopting BiLSTM, attention and CRF components, looks similar to previous sequence tagging systems including BiLSTM (Huang et al., 2015) and OpenTag (Zheng et al., 2018). But in fact our model is fundamentally different from previous works: unlike their strategy to regard attribute as only tag, we model attribute semantically, capture its semantic interaction with title via attention mechanism, then generate attribute-comprehension title representation to CRF for final tagging. Next we will describe the architecture of our model in detail.

Word Representation Layer. We map each word in the title and attribute to a high-dimensional vector space through the pre-trained Bidirectional

Encoder Representations from Transformers (BERT) (Devlin et al., 2018) which is the state-of-the-art language representation model. For each word in a sentence, BERT generates a particular word representation which considers the specific contexts. Formally, BERT encodes the title T and attribute A into a sequence of word representations $\{w_1^t, w_2^t, \ldots, w_m^t\}$ and $\{w_1^a, w_2^a, \ldots, w_n^a\}$.

Contextual Embedding Layer. Long-Short Term Memory (LSTM) Neural Network (Hochreiter and Schmidhuber, 1997) addresses the vanishing gradient problems and is capable of modeling long-term contextual information along the sequence. Bidirectional LSTM (BiLSTM) captures the context from both past and future time steps jointly while vanilla LSTM only considers the contextual information from the past.

In this work, we adopt two BiLSTMs to model the title and attribute representation individually. One BiLSTM is used to get hidden states as contextual representation of title $\mathbf{H^t} = \{h_1^t, h_2^t, \ldots, h_m^t\}$.

$$h_i^t = \left[\overrightarrow{h_i^t}; \overleftarrow{h_i^t}\right] = \text{BiLSTM}\left(\overrightarrow{h_{i+1}^t}, \overleftarrow{h_{i-1}^t}, \mathrm{w_i^t}\right)$$

Another BiLSTM is used to obtain the attribute representation. Slightly different from the design for title, we only use the last hidden state of BiLSTM as the attribute representation h_a since the length of attribute is normally much shorter (i.e., no more than 5).

$$h^a = \left[\overrightarrow{h_n^a}; \overleftarrow{h_n^a}\right] = \text{BiLSTM}\left(\overrightarrow{h_n^a}, \overleftarrow{h_n^a}, \mathrm{w_n^a}\right)$$

Attention Layer. In Natural Language Processing (NLP), attention mechanism was first used in Neural Machine Translation (NMT) (Bahdanau et al., 2014) and has achieved a great success. It is designed to highlight the important information in a sequence, instead of paying attention to everything.

OpenTag (Zheng et al., 2018) uses self-attention (Vaswani et al., 2017) to capture the important tokens in the title, but treats attribute only as a type and neglects attribute semantic information. Thus, OpenTag has to introduce one set of tags (B_a, I_a) for each attribute a, leading to its failure to be applicable in e-Commerce which has ten of thousands attributes. Different from their work, our model takes the hidden semantic interaction between attribute and title into consideration by computing the similarities between the attribute and each word in title. This means different tokens in the title would be attended in order to extract values for different attributes, resulting in different weight matrix. Thus, our model is able to handle huge amounts of attributes with only one set of tags (B, I, O). Even for attributes that have never been seen before, our model is able to identify tokens associated with it from the title by modeling its semantic information.

We first compute the similarity between the attribute and each word in title to obtain attention vector $\mathrm{S} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. The attribute-comprehension title is $\mathbf{C} = \mathrm{S} \odot \mathbf{H^t}$, where $\odot$ represents element-wise. This vector indicates the weighted sum of words in the title with respect to the attribute. The similarity function between two vectors is measured by *cosine* similarity:

$$\alpha_i = cosine\left(h_i^t, h^a\right)$$

Output Layer. The goal of this task is to predict a tag sequence that marks the position of attribute values in the title. CRF is often used in sequence tagging model because it captures dependency between the output tags in a neighborhood. For example, if we already know the tag of a token is *I*, this decreases the probability of the next token to be *B*.

We concatenate the title $\mathbf{H^t}$ and attribute-comprehension title $\mathbf{C}$ to obtain a matrix $\mathbf{M} = [\mathbf{H^t}; \mathbf{C}]$, which is passed into the CRF layer to predict tag sequence. Each column vector of $\mathbf{M}$ expected to contain contextual information about the word with respect to the title and attribute. The joint probability distribution of tags y is given by:

$$Pr\left(y|T;\psi\right) \propto \prod_{i=1}^{m} exp\left(\sum_{k=1}^{K} \psi_k f_k\left(y_{i-1}, y_i, \boldsymbol{M_i}\right)\right)$$

where ψ_k is corresponding weight, f_k is the feature function, K is the number of features. The final output is the best label sequence y^* with the highest conditional probability:

$$y^* = argmax_y Pr\left(y|u;\psi\right)$$

Training. For training this network, we use the maximum conditional likelihood estimation:

$$L\left(\psi\right) = \sum_{i=1}^{N} Pr\left(y_i|u_i;\psi\right)$$

where N is the number of training instances.

Groups	Occurrence	# of Attributes	Example of attributes
High	[10,000, ∞)	10	Gender, Brand Name, Model Number, Type, Material
Sub-high	[1000, 10,000)	60	Feature, Color, Category, Fit, Capacity
Medium	[100, 1000)	248	Lenses Color, Pattern, Fuel, Design, Application
Low	[10, 100)	938	Heel, Shaft, Sleeve Style, Speed, Carbon Yarn
Rare	[1, 10)	7,650	Tension, Astronomy, Helmet Light, Flashlight Pouch

Table 1: The statistics and examples of $8,906$ attributes with different frequencies in dataset AE-650K.

4 Experimental Setup

4.1 Dataset

We use $344,373$ products collected from AliExpress Sports & Entertainment category as our dataset. For each product, their attributes and corresponding values presented in Item Specific are retained as ground truth for evaluation. The number of attributes varies greatly from different products. For example, up to 85 attributes are listed in one GQBQ children sport shoes product[2]. On average, each product contains about 10 attributes. We pair product title with its attributes and values present in Item Specific to form $3,383,547$ triples, i.e., {*title*, *attribute*, *value*} as initial dataset.

In initial dataset, there are $513,564$ positive triples (15%) whose value is included in title, the remainder are negative triples whose value is marked as 'NULL' as it is missing in title. We randomly select $143,846$ negative triples, then combine them with all positive triples to compose the dataset AE-650K whose positive-negative ratio is 4:1. Then this set of $657,410$ triples is partitioned into training, development and test set with the ratio of 7:1:2. In total, the AE-650k dataset contains $8,906$ types of attributes and their distributions are extremely uneven. In order to have a deep insight into the attribute distribution, we categorize them into five groups (i.e., High, Sub-high, Medium, Low and Rare frequency) according their occurrences. Table 1 shows the number of unique attributes in each frequency group together with some examples. We observe that high frequency attributes are more general (e.g., *Gender*, *Material*), while low and rare frequency attributes are more product specific (e.g., *Sleeve Style*, *Astronomy*). For example, one Barlow lens product has value 'Telescope Eyepiece for *Astronomy* [3]. In addition, we find these attributes has "long tail" phenomenon, that is, a small number of general attributes can basically define a product while there are a large number of specific attributes to define products more detailedly. These details are important in the accurate produces recommendation or other personalized services.

Attributes	Train	Dev	Test
Brand Name	50,413	5,601	14,055
Material	22,814	2,534	6,355
Color	5,594	621	1,649
Category	5,906	590	1,462
Total	84,727	9,346	23,521

Table 2: Statistics of dataset AE-110K.

In order to make fair comparison between our model and previous sequence tagging models which cannot handle huge amounts of attributes, we pick up the four frequent attributes (i.e., *Brand Name*, *Material*, *Color* and *Category*) to compose the second dataset AE-110k with a total of $117,594$ triples. Table 2 shows the statistics and distributions of attributes in AE-110k.

Moreover, since the dataset is automatically constructed based on *Exact Match* criteria by pairing product title with its attributes and values present in Item Specific, it may involve some noises for positive triples. For example, the title of a '*dress*' contains 'long dresses', the word 'long' may be tagged as values for attributes *Sleeve Length* and *Dresses Length* simultaneously. Thus we randomly sampled $1,500$ triples from AE-650k for manual evaluation and the accuracy of automatic labeling is 95.6%. This shows that the dataset is high-quality.

[2]https://www.aliexpress.com/item/32956754932.html

[3]https://www.aliexpress.com/item/32735772355.html

4.2 Evaluation Metrics

We use precision, recall and F_1 score as evaluation metrics denoted as P, R and F_1. We follow *Exact Match* criteria in which the full sequence of extracted value need to be correct. Clearly, this is a strict criteria as one example gets credit only when the tag of each word is correct.

4.3 Baselines

To make the comparison reliable and reasonable, three sequence tagging models serve as baselines due to their reported superior tagging results like OpenTag (Zheng et al., 2018) or their typical representation (Huang et al., 2015).

- **BiLSTM** uses the pre-trained BERT model to represent each word in title, then applies BiLSTM to produce title contextual embedding. Finally, a *softmax* function is exploited to predict the tag for each word.

- **BiLSTM-CRF**(Huang et al., 2015) is considered to be the pioneer and the state-of-the-art sequence tagging model for NER which uses CRF to model the association of predicted tags. In this baseline, the hidden states generated by BiLSTM are used as input features for CRF layer.

- **OpenTag**(Zheng et al., 2018) is the recent sequence tagging model for this task which adds self-attention mechanism to highlight important information before CRF layer. Since the source code of OpenTag is not available, we implement it using Keras.

4.4 Implementation Details

All models are implemented with Tensorflow (Abadi et al., 2016) and Keras (Chollet et al., 2015). Optimization is performed using Adam (Kingma and Ba, 2014) with default parameters. We train up to 20 epochs for each model. The model that performs the best on the development set is then used for the evaluation on the test set. For all models, the word embeddings are pre-trained via BERT and the dimension is 768. The dimension of the hidden states in BiLSTM is set to 512 and the minibatch size is fixed to 256. The *BIO* tagging strategy is adopted. Note that only one global set of *BIO* tags for any attributes is used in this work.

Attributes	Models	P (%)	R (%)	F_1 (%)
Brand Name	BiLSTM	95.08	96.81	95.94
	BiLSTM-CRF	95.45	97.17	96.30
	OpenTag	95.18	97.55	96.35
	Our model-110k	**97.21**	96.68	**96.94**
	Our model-650k	**96.94**	**97.14**	**97.04**
Material	BiLSTM	78.26	78.54	78.40
	BiLSTM-CRF	77.15	78.12	77.63
	Opentag	78.69	78.62	78.65
	Our model-110k	**82.76**	**83.57**	**83.16**
	Our model-650k	**83.30**	**82.94**	**83.12**
Color	BiLSTM	68.08	68.00	68.04
	BiLSTM-CRF	68.13	67.46	67.79
	Opentag	71.19	70.50	70.84
	Our model-110k	**75.11**	**72.61**	**73.84**
	Our model-650k	**77.55**	**72.80**	**75.10**
Category	BiLSTM	82.74	78.40	80.51
	BiLSTM-CRF	81.57	79.94	80.75
	Opentag	82.74	80.63	81.67
	Our model-110k	**84.11**	**80.80**	**82.42**
	Our model-650k	**88.11**	**81.79**	**84.83**

Table 3: Performance comparison between our model and three baselines on four frequent attributes. For baselines, only the performance on `AE-110K` is reported since they do not scale up to large set of attributes; while for our model, the performances on both `AE-110K` and `AE-650K` are reported.

5 Results and Discussion

We conduct a series of experiments under various settings with the purposes to (1) make comparison of attribute extraction performance on frequent attributes with existing state-of-the-art models; (2) explore the scalability of our model up to thousands of attributes; and (3) examine the capability of our model in discovering new attributes which have not been seen before.

5.1 Results on Frequent Attributes

The first experiment is conducted on four frequent attributes (i.e., with sufficient data) on `AE-110k` and `AE-650k` datasets. Table 3 reports the comparison results of our two models (on `AE-110k` and `AE-650k` datasets) and three baselines. It is observed that our models are consistently ranked the best over all competing baselines. This indicates that our idea of regarding 'attribute' as 'query' successfully models the semantic information embedded in attribute which has been ignored by previous sequence tagging models. Besides, different from the self-attention mechanism only in-

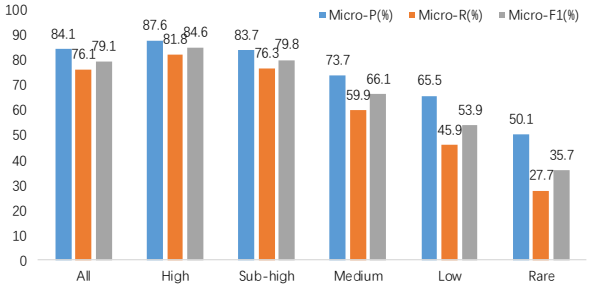

Figure 3: Performance of our model on $8,906$ attributes in AE-650K dataset. 'All' stands for all attributes while 'High', 'Sub-high', 'Medium', 'Low' and 'Rare' denote the five frequency groups of attributes defined in Table 1, respectively.

side title adopted by OpenTag, our interacted similarity between attribute and title does attend to words which are more relevant to current extraction.

In addition, our model is the only one that can be applied to AE-650K dataset which contains $8,906$ types of attributes. From Table 3, we compare the performance of our two models trained on different sizes of triples. It is interesting to find that extra training data on other attributes boosts the performances of the target four attributes, and outperforms the best baseline by average 3% in F_1 score. We believe the main reason is that all the other attributes in AE-650k can be viewed as relevant tasks from Multi-task (Caruana, 1997) perspective. Usually, the model would take the risk of over-fitting if it is only optimized upon the target attributes due to unavoidable noises in the dataset. However, the Multi-task learning implicitly increases training data of other relevant tasks having different noise patterns and can average these noise patterns to obtain a more general representation and thus improve generalization of the model.

5.2 Results on Thousands of Attributes

The second experiment is to explore the scalability of models up to thousands of attributes. Clearly, previous sequence tagging models fail to report results on large amounts of tags for attributes. Using a single model to handle large amounts of attributes is one advantage of our model. To verify this characteristic, we compute *Micro-P*, *Micro-R*, *Micro-*F_1 on entire test set of AE-650k, as shown in the leftmost set of columns of Figure 3. The performances of our model on $8,906$ attributes reach 84.13%, 76.08% and 79.12%, respectively.

Attributes	P (%)	R (%)	F_1 (%)
Frame Color	63.16	48.00	54.55
Lenses Color	64.29	40.91	50.00
Shell Material	54.05	44.44	48.78
Wheel Material	70.59	37.50	48.98
Product Type	64.86	43.29	51.92

Table 4: Performance of our model in discovering values for new attributes.

In order to validate the robustness of our model, we also perform experiments on five attribute frequency groups defined in Table 1. Their results are shown in Figure 3. We observe that our model achieves *Micro-*F_1 of 84.60% and 79.79% for frequent attributes in 'High' and 'Sub-high' groups respectively. But more importantly, our model achieves good performance (i.e., *Micro-*F_1 66.06% and 53.94% respectively) for less frequent attributes in 'Medium' and 'Low' groups, and even a promising result (i.e., *Micro-*F_1 35.70%) for 'Rare' attributes which are presented less than 10 times. Thus, we are confident to conclude that our model has the ability to handle large amounts of attributes with only a single model.

5.3 Results of Discovering New Attributes

To further examine the ability of our model in discovering new attributes which has never been seen before, we select 5 attributes with relatively low occurrences: *Frame Color*, *Lenses Color*, *Shell Material*, *Wheel Material*, and *Product Type*. We shuffle the AE-650K dataset to make sure they are not in training and development set, and evaluate the performance for these 5 attributes. Table 4 reports the results of discovering 5 new attributes. It is not surprising to see that our model still achieves acceptable performance (i.e., averaged F_1 50.85%) on new attributes with no additional training data. We believe that some data in training set are semantically related to unseen attributes and they provide hints to help the extraction.

To further confirm this hypothesis, we map attributes features h^a generated by contextual embedding layer into two-dimensional space by t-SNE (Rauber et al., 2016), as shown in Figure 4. In Figure 4 the four colors of circles represent the attributes of *Color*-related,[4] *Type*-related, *Materi-*

[4] '*a*-related' denotes all attributes whose text contains the substring *a*.

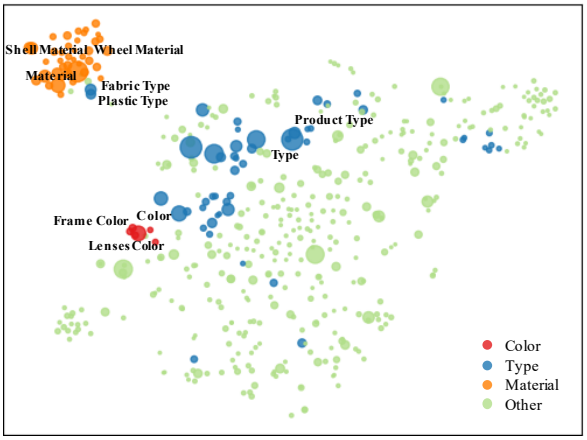

Figure 4: Distribution between semantically related new and existing attributes. E.g., *Shell Material* and *Wheel Material* are new attributes while *Material* is frequently known attributes.

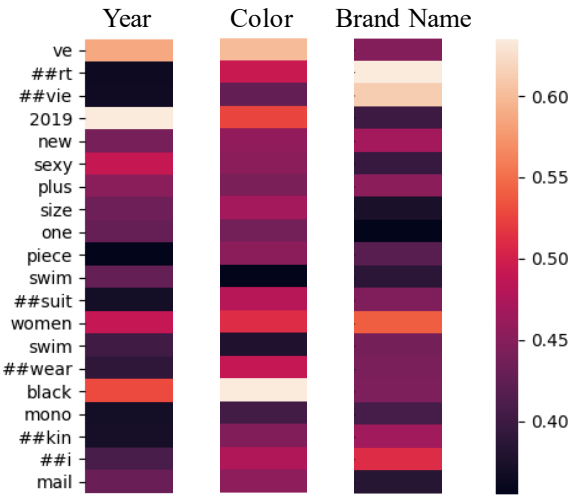

Figure 5: The heat map of attention vector S.

al-related and others respectively, and the areas are proportional to the frequency of attributes. An interesting observation is that *Color*-related and *Material*-related attributes are clustered into a small and concentrated area of two-dimensional space, respectively. Meanwhile, although *Type* and *Product Type* are very close, the distribution of all *Type*-related attributes is scattered in general. It may be because *Type* is not a specifically defined concept compared to *Color* or *Material*, the meaning of a *Type*-related attribute is determined by the word paired with *Type*. Therefore, we select two *Type*-related attributes adjacent to *Material* and find they are *Fabric Type* and *Plastic Type*. In fact, these two attributes are indeed relevant to the material of products.

To verify the ability of our model to handle a larger number of new attributes, we collect additional $20,532$ products from new category `Christmas`, and form $46,299$ triples as test set. The `Christmas` test set contains $1,121$ types of attributes, 708 of which are new attributes. Our model achieves *Micro*-F_1 of 66.37% on this test set. This proves that our model has good generalization and is able to transfer to other domains with a large number of new attributes.

5.4 Attention Visualizations

To illustrate the attention learned from the product in Figure 1, we plot the heat map of attention vectors S for three attributes (*Year*, *Color* and *Brand Name*) where the lighter the color is the higher the weight is. Since each bar in the heat map represents the importance of a word in the title of each attribute, it indirectly affects the prediction decision. By observing Figure 5, we see that our model indeed adjusts the attention vector according to different attributes to highlight the value.

6 Related Work

Previous work for attribute value extraction use rule-based extraction techniques (Vandic et al., 2012; Gopalakrishnan et al., 2012) which use domain-specific seed dictionary to spot key phrase. Ghani et al. (2006) predefine a set of product attributes and utilize supervised learning method to extract the corresponding attributes values. An NER system was proposed by Putthividhya and Hu (2011) for extracting product attributes and values. In this work, supervised NER and bootstrapping technology are combined to expand the seed dictionary of attribute values. However, these methods suffer from Limited World Assumption. More (2016) build a similar NER system which leverage existing values to tag new values.

With the development of deep neural network, several different neural network methods have been proposed and applied in sequence tagging successfully. Huang et al. (2015) is the first to apply BiLSTM-CRF model to sequence tagging task, but this work employ heavy feature engineering to extract character-level features. Lample et al. (2016) 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. Convolutional neural network (CNN) (Le-

Cun et al., 1989) is employed to model character-level information in Chiu and Nichols (2016) which achieves competitive performance for two sequence tagging tasks at that time. Ma and Hovy (2016) propose an end to end LSTM-CNNs-CRF model.

Recently, several approaches employ sequence tagging model for attribute value extraction. Kozareva et al. (2016) adopt BiLSTM-CRF model to tag several product attributes from search queries with hand-crafted features. Furthermore, Zheng et al. (2018) propose an end-to-end tagging model utilizing BiLSTM, CRF, and Attention without any dictionary and hand-crafted features. Besides extracting attribute value from title, other related tasks have been defined. Nguyen et al. (2011); Sheth et al. (2017); Qiu et al. (2015) extracted attribute-value pairs from specific product description.

7 Conclusion

To extract product attribute values in e-Commerce domain, previous sequence tagging models face two challenges, i.e., the huge amounts of product attributes and the emerging new attributes and new values that have not been seen before. To tackle the above issues, we present a novel architecture of sequence tagging with the integration of attributes semantically. Even if the attribute size reaches tens of thousands or even millions, our approach only trains a single model for all attributes instead of building one specific model for each attribute. When labeling new attributes that have not encountered before, by leveraging the learned information from existing attributes which have similar semantic distribution as the new ones, this model is able to extract the new values for new attributes. Experiments on a large dataset prove that this model is able to scale up to thousands of attributes, and outperforms state-of-the-art NER tagging models.

Acknowledgements

The authors wish to thank all reviewers for their helpful comments and suggestions. This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program. This work has been completed during Huimin Xu and Xin Mao's internship in Alibaba Group.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016.*, pages 265–283.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Daniel M. Bikel, Richard M. Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

François Chollet et al. 2015. Keras. `https://keras.io`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew E. Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explorations*, 8(1):41–48.

Vishrawas Gopalakrishnan, Suresh Parthasarathy Iyengar, Amit Madaan, Rajeev Rastogi, and Srinivasan H. Sengamedu. 2012. Matching product titles using web-based enrichment. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 605–614.

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

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289.

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

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Xuezhe Ma and Eduard 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 (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *CoRR*, abs/1608.04670.

Hoa Nguyen, Ariel Fuxman, Stelios Paparizos, Juliana Freire, and Rakesh Agrawal. 2011. Synthesizing products for online catalogs. *PVLDB*, 4(7):409–418.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1557–1567.

Disheng Qiu, Luciano Barbosa, Xin Luna Dong, Yanyan Shen, and Divesh Srivastava. 2015. DEXTER: large-scale discovery and extraction of product specifications on the web. *PVLDB*, 8(13):2194–2205.

Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing time-dependent data using dynamic t-sne. In *Eurographics Conference on Visualization, EuroVis 2016, Short Papers, Groningen, The Netherlands, 6-10 June 2016.*, pages 73–77.

Amit P. Sheth, Axel Ngonga, Yin Wang, Elizabeth Chang, Dominik Slezak, Bogdan Franczyk, Rainer Alt, Xiaohui Tao, and Rainer Unland, editors. 2017. *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*. ACM.

Damir Vandic, Jan-Willem van Dam, and Flavius Frasincar. 2012. Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

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.