Learning Cross-Task Attribute - Attribute Similarity for Multi-task Attribute-Value Extraction

Mayank Jain [*], Sourangshu Bhattacharya[*], Harshit Jain[**], Karimulla Shaik[**], and Muthusamy Chelliah[**]

[*]Computer Science & Engineering, IIT Kharagpur, India.
jmayank@gmail.com, sourangshu@iitkgp.ac.in
[**]Flipkart India Pvt. Ltd., Bengaluru, India.
{ harshit.jain,karimulla.shaik,muthusamy.c}@flipkart.com,

Abstract

Automatic extraction of product attribute-value pairs from unstructured text like product descriptions is an important problem for e-commerce companies. The attribute schema typically varies from one category of products (which will be referred as vertical) to another. This leads to extreme annotation efforts for training of supervised deep sequence labeling models such as LSTM-CRF, and consequently not enough labeled data for some vertical-attribute pairs. In this work, we propose a technique for alleviating this problem by using annotated data from related verticals in a multi-task learning framework. Our approach relies on availability of similar attributes (labels) in another related vertical. Our model jointly learns the similarity between attributes of the two verticals along with the model parameters for the sequence tagging model. The main advantage of our approach is that it does not need any prior annotation of attribute similarity. Our system has been tested with datasets of size more than 10000 from a large e-commerce company in India. We perform detailed experiments to show that our method indeed increases the macro-F1 scores for attribute value extraction in general, and for labels with low training data in particular. We also report top labels from other verticals that contribute towards learning of particular labels.

1 Introduction

Online e-commerce marketplaces (e.g., Flipkart) operate by efficiently matching customer queries and browsing habits to appropriate seller inventory. Inventory is stored in a catalog which consists of images, structured attributes (key-value pairs) and unstructured textual description as shown in figure 1. Products of same kind (e.g., digital camera) are thus described using a unique set of attributes (e.g., zoom, resolution) – helping

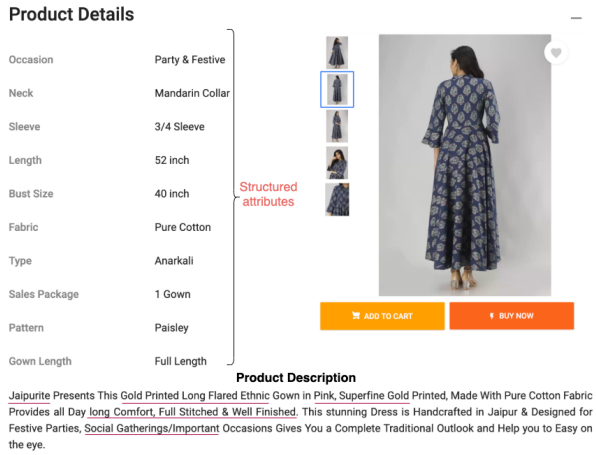


Figure 1: **A snapshot of structured attributes and product description - underlined words wherein is important, additional information not provided by seller in attributes.**

faceted navigation, merchandizing, search ranking and comparative summary.

Onboarding products in a catalog requires populating the structured as well as unstructured parts. The time a seller has to spend on a product addition request is proportional to the quantum of information that he/she has to provide. On the other hand, correctness and completeness of catalog results in better product discovery, leading to a trade-off with its onboarding time. A good amount of attributes information is present in product description as well. This motivates us to extract the information from unstructured text instead of explicitly asking sellers for attributes. Additional information in description (e.g., precise features, relation between products) as shown in figure 1 helps to enrich the catalog as well. The extracted attributes can be used to check consistency between unstructured and structured data provided by seller and thus quality control of addition request.

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 79–87

We design supervised deep learning techniques for the problem of attribute value extraction. Figure 2, shows a typical input sentence and the corresponding `B, I, O` tags. The task of our model is to predict the tags given an input sentence. This problem is related supervised sequence labelling problem (Zheng et al., 2018; Lample et al., 2016). However, this technique needs a lot of training data points (sentence - label pairs) to perform effectively, which in turn requires massive annotation efforts on the part of e-commerce companies - reduction of which is an ongoing challenge; OpenTag (Zheng et al., 2018) uses active learning to annotate only most informative examples.

E-commerce companies however have the products categorized as different verticals, e.g. *dress*, *jeans*, etc. Each of these verticals have a different set of attributes, and hence needs to annotated using different models. A lot of the attributes among these verticals are common, or related though. Hence, it should be possible to borrow information from annotations given in different verticals, to improve the prediction performance of a given vertical. The only challenge is that correspondences between similar labels of different verticals is not available readily.

Our main contribution here is thus to develop a multi-task learning (MTL) model (Ruder, 2017) which can simultaneously learn attribute extraction and attribute-attribute similarity for multiple verticals (here we report with only two verticals at a time). We do so by using a soft coupling loss function across pairs of similar (context,label) combinations between the two tasks, where similarity is learned using attention mechanism. The naive version of such an objective will be prohibitively large to optimize. We propose to use a cosine similarity based shortlist, which makes the solution feasible.

We validate our method using a large corpus (more than 10000 product descriptions, across 6 verticals) collected from the e-commerce company - Flipkart. Extensive experimentation shows that our method improves performance of prediction on almost all the verticals, and especially shows upto 50% improvement for many labels which have low number of training examples. This is especially interesting since we find that number of instances with an attribute is highly skewed across the attributes. Detailed analysis also confirms that the attention mechanism indeed discovers similar attributes from other verticals to borrow information from.

Words	Black	levis	jeans	for	mens	with	cotton	blend	fabric
Attributes	B color	B brand	O	O	B Ideal_for	O	B Fabric	I Fabric	O

Figure 2: Sample tagged data from Jean Vertical.

2 Related Work

Attribute extraction: Various tokens (e.g., Apple) in an offer title are classified into attribute names (e.g., brand) relevant to the product (e.g., smartphone) (Joshi et al., 2015). For recognizing attributes (e.g., product family) in a short text segment, missing KB entries are leveraged through word embeddings learned on an unlabeled corpus (Kozareva et al., 2016). (Joshi et al., 2015) investigates whether distributed word vectors benefit NER in the e-commerce domain where entities are item properties (e.g., brand name, color, material, clothing size). (Xu et al., 2019) regards each attribute as a query and adopts only one global set of BIO tags for any attribute to reduce the burden of attribute tag or model explosion. OpenTag (Zheng et al., 2018) uses active learning along with a deep tagging model to update a product catalog with missing values for many attributes of interest from seller-provided title/description. To create the initial labeled data set, (Rezk et al., 2019) proposes bootstraping of seed data by extracting new values from unstructured text in a domain/language-independent fashion. Through category conditional self-attention and multi-task learning, a knowledge extraction model Attribute prediction and value extraction tasks are jointly modelled (Zhu et al., 2020) from multiple aspects towards interactions between attributes and values. Contrastive entity linkage (Embar et al., 2020) helps identify grocery product attribute pairs that share same value (e.g., brand, manufacturer, product line) and differ from each other (e.g., package size, color). Retailers do not always provide clean data as textual descriptions in product catalog (e.g., non-distinctive names (cotton, black t-shirt), blurred distinction (Amazon is a product/vs. brand), homonyms (Apple)). (Alonso et al., 2019) discovers such attribute relationships towards a brand-product knowledge graph from diverse input data sources.

Multi-task Learning (MTL): Significant theoretical interest exists in MTL since it offers excel-

lent generalization performance in domains where training data is scarce (Maurer et al., 2016). In NLP, (Collobert and Weston, 2008) proposed a unified deep learning architecture for many common tasks e.g. POS tagging, chunking, etc. (Yang and Hospedales, 2017) presented a new representation MTL framework that learns cross-task sharing structure at every layer in a deep network. (Rijula Kar, 2018) proposed a task-sensitive representation learning framework that learns mention-dependent representations for NED, and violates norm to share parameters in the final layer.

(Wang et al., 2020) treats each attribute as a question finding the best answer span corresponding to its value in product context - modelled by a BERT encoder shared across all attributes for scalability. A distilled masked language model improving generalizability is then integrated with the encoder into a unified MTL framework. (Karamanolakis et al., 2020) applies to thousands of product categories organized in a hierarchical taxonomy. However, existing methods do not automatically discover attribute-attribute similarity from data, without taking attribute hierarchy as input.

3 Methods

In this section, we describe a novel multi-task approach to improving the accuracy of a supervised attribute value extraction system. We start with the attribute-value extraction system, based on deep bidirectional LSTM model, described in OpenTag (Zheng et al., 2018). Our main idea here is to leverage the information contained in instances of related tasks, e.g. in our case related domains / verticals of products. The key challenge in our case is that the set of labels across verticals need not be same, or even *aligned*. For example, the label `PROCESSOR_TYPE` is a valid label for `LAPTOP` vertical but does not make sense for `DRESS` vertical. On the other hand, the set of values for the common label `BRAND` will be very different for the vertical `DRESS` compared to the vertical `LAPTOP`. Hence, our core challenge here is to determine the similarities between labels automatically in the context of each vertical in order to leverage the information from a related vertical. The proposed architecture is described in figure 3.

3.1 Problem setup

Each instance of the (single-task) attribute-value extraction problem comes with an input sentence denoted by a sequence of words $\mathbf{w} = \{w_1, \ldots, w_n\}$ and a corresponding set of labels $\mathbf{y} = \{y_1, \ldots, y_n\}$. The task is to design a supervised ML algorithm which given the input sentence $\mathbf{w}$, predicts the output labels $\mathbf{y}$. Here, the labels correspond to the attributes, e.g. `COLOR`, and words correspond to the predicted values. Following common practice, we use 3 types of labels (also called tags): `B, I, O`. Here `B` and `I` are prepended to the label to indicate begining and end of a multi-word tag, respectively, while `O` refers to no tag for the word. For example, the multi-word color "light green" may be tagged as `B_COLOR` and `I_COLOR`.

This is an instance of sequence labeling problem (Lample et al., 2016), and the LSTM-CRF model proposed by Lample et al. (Lample et al., 2016) is the a state of the art model for this task. For each word w_i, we obtain the corresponding word embedding x_i using a concatenation of its glove embedding (Pennington et al., 2014) and it's character based embedding. The word embeddings of a sentence $\mathbf{x} = \{x_1, \ldots, x_n\}$ is passed through a Bidirectional LSTM (BiLSTM) layer to produce the context sensitive word embedding $\mathbf{h}$:

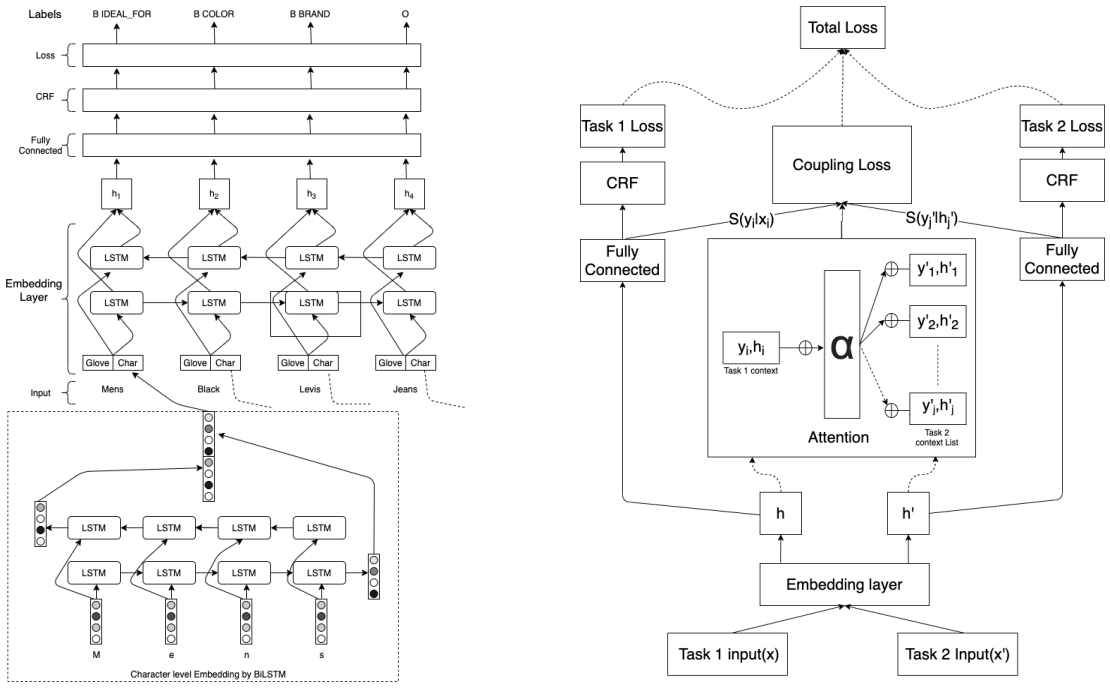
$$\mathbf{h} = \text{BiLSTM}(\mathbf{x}) \tag{1}$$

We call this the the embedding layer for our input which is common to both single and multi-task models. Figure 3(a) describes the architecture in detail.

For the multi-task attribute-value extraction problem, the input is a sentence $w_j^t j = 1, \ldots, n$, and the output of model is a sequence of labels $y_j^t, j = 1, \ldots, n$, where $t = 1, \ldots, T$. In this paper we only consider the setting of $T = 2$, i.e. we learn from 2 tasks at a time, due to scalability reasons. However, in theory our method can be extended to learning from more than 2 tasks. We compute the word embeddings $\mathbf{x}$ and context dependent word embeddings $\mathbf{h}$ in a similar manner as described above.

3.2 Single-task attribute-value extraction

We use the LSTM-CRF model with character embeddings (Lample et al., 2016; Zheng et al., 2018) as our baseline single task model. For a given input sentence the word embeddings $\mathbf{x}$ and the con-



(a) (b)

Figure 3: (a) Architecture of single task model showing the sentence embedding layers (b) High-level architecture of the multi-task attribute-value extraction model

text sensitive word embeddings $\mathbf{h}$ are computed as described above. The context sensitive word embeddings $h_i, i = 1, \ldots, n$ is then passed through a fully connected layer to produce the score $s(y)$ for every possible label y. This is parameterized by the matrix $W \in R^{d \times k}$ and $b \in R^k$ where d is the dimension of h_i and k is the total number of possible labels. Hence the score vector for every label is computed as:

$$s_i(.|\mathbf{x}) = W \times h_i + b \forall i = 1, \ldots, n$$

where n is the length of sentence. We can interpret the k^{th} component of s_i, denoted as $s_i(y = k|h_i)$, as the score of class k for word w_i . Now, given a sequence of words vectors $\mathbf{x}$, a sequence of score vectors $\{s_1(y|\mathbf{x}), \ldots, s_n(y|\mathbf{x})\}$, and a sequence of keys $\mathbf{y}$, a linear-chain CRF defines a global score $C \in R$ as,

$$C(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} s_i(y_i|\mathbf{x}) + \sum_{i=1}^{n-1} T(y_i, y_{i+1}|\mathbf{x})$$

Here, $s(y|\mathbf{x})$ is the y^{th} component of the s vector and $T(y, y')$ is the transition score from label y to y', which is used to capture label dependency.

A softmax over all possible tag sequences yields a probability for the sequence $\mathbf{y}$. $P(\mathbf{y}|\mathbf{x}) = \frac{e^{C(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{y}' \in \mathcal{Y}} e^{C(\mathbf{x},\mathbf{y}')}}$ During training, we maximize the log-probability of the correct key sequence: $log(P(\mathbf{y}|\mathbf{x})) = C(\mathbf{x}, \mathbf{y}) - log(\sum_{\mathbf{y}' \in \mathcal{Y}} e^{C(\mathbf{x},\mathbf{y}')})$ Here $\mathcal{Y}$ is the set of all possible labellings for sequence $\mathbf{x}$. Given a dataset of sequences and labels $\mathcal{D} = \{(\mathbf{x}_j, \mathbf{y}_j), j = 1, \ldots, m$, we can define the CRF loss as the negative log-likelihood:

$$L_{CRF}(W, b) = \sum_{j=1}^{m} -log(P(\mathbf{y}_j|\mathbf{x}_j))$$

(Lample et al., 2016) describes a method for learning the model parameters and inferring the partition function and scores by minimizing the above objective w.r.t. W and b.

3.3 Multi-task attribute-value extraction

As mentioned above, for multi-task attribute-value extraction, we have sequence and label combinations $(\mathbf{x}^t, \mathbf{y}^t)$ for two tasks, $t \in \{1, 2\}$. We also note that we have a common set of embedding layers (both word representation and BiLSTM) for the two tasks. However, the feedforward layer used for scoring the labels are specific to the tasks. Hence:

$$s_i^t(.|\mathbf{x}) = W^t \times h_i + b^t, \forall i = 1, \ldots, n\,; \forall t = \{1, 2\}$$

The score and loss functions can be defined analogously to the single task model as: $C^t(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} s_i^t(y_i|\mathbf{x}) + \sum_{i=1}^{n-1} T^t(y_i, y_{i+1}|\mathbf{x})$, and $log(P^t(\mathbf{y}|\mathbf{x})) = C^t(\mathbf{x}, \mathbf{y}) - log(\sum_{\mathbf{y}' \in \mathcal{Y}} e^{C^t(\mathbf{x}, \mathbf{y}')})$. Given the multi-task dataset $\mathcal{D}^t = \{(\mathbf{x}_j^t, \mathbf{y}_j^t), j = 1, \ldots, m^t,\ t = \{1, 2\}$, our loss function can be written as:

$$L_{CRF}(W, b) = \sum_{t=1}^{2} \sum_{j=1}^{m^t} -log(P^t(\mathbf{y}_j^t|\mathbf{x}_j^t))$$

Hence, only parameters of the embedding layers get affected by the multi-task paradigm here, since those are the only shared layers between the tasks. However, these parameters are independent of the labels and are thus relatively robustly learned by just using a reasonably large corpus of input sentences. Another mechanism for borrowing information between tasks is through "soft coupling" (Ruder, 2017) of various scores or parameters which are not explicitly shared. In the next section, we devise a soft coupling loss between instances of the two tasks which achieve transfer of information at the granularity of labels.

3.4 Coupling loss

The principle we use for coupling of scores $s_i^t(y|x)$ is: *similar labels in similar contexts should have similar scores*. Recall that the dataset for multi-task attribute value extraction consists of two sets of instances $\mathcal{D}^1$ and $\mathcal{D}^2$, for each of the two tasks. Since we are attempting to compare the model predictions for the two tasks, the coupling loss depends on two *contexts*, one from each task: $(\mathbf{x}_j, \mathbf{y}_j, i)$ and $(\mathbf{x}_{j'}, \mathbf{y}_{j'}, i')$. Here, j and j' denotes indices of instances for the two tasks, and i and i' indices within the each sentence instance to the two tasks. We note that since the are ~ 1000 instances for each task, and ~ 10 length sentences for each instance, the total number of terms for this loss will be $\sim 10^8$ $((10 \times 1000)^2)$. This is prohibitively large for our training purpose, and also is wasteful, since not all *contexts* (combination of instance j and position i) are related to each other.

Hence, as a first step we create a shortlist of pairs of contexts $((i, j), (i', j'))$ which can borrow informations from each other, by thresholding on the cosine similarity between the a windows around the contexts $u_{i,j}$ and $u'_{i',j'}$:

$$L = \{((i, j), (i', j')) \mid \text{cosine\_sim}(u_{i,j}, u'_{i',j'}) > \text{thresh}\}$$

Here, note that $u(i, j)$ is the word embedding of a window around the context (i, j).

Context coupling error: Our next challenge is to design a mechanism to figure out similar contexts and similar labels. We use the softmax attention mechanism to automatically learn the similar label-context combinations, simultaneously as we also learn the scoring function. For efficiency of parameters, we use the Luong attention. Hence the attention score for context (i, j) from task 1 over context (i', j') from task 2 is given by:

$$A(j, i, j', i') = \frac{e^{\alpha(j,i,j',i')}}{\sum_{(\hat{j},\hat{i}) \in L(j,i)} e^{\alpha(j,i,\hat{j},\hat{i})}}$$
$$\alpha(j, i, j', i') = u(i, j)^T \text{diag}(\mathbf{a}) u(i', j')$$

Here, $\mathbf{a} = (a_1, ..., a_d)$ are learnable parameters of same dimension as the word embeddings, and $L(j, i) = \{(j', i') | ((i, j), (i', j')) \in L\}$. The *context-coupling error* is defined as:

$$CCE(L, \mathbf{a}) = \sum_{((j,i),(j',i')) \in L} A(j, i, j', i') \times |(s_i(y_{ji}|\mathbf{x}_j) - s'_{j'}(y'_{j',i'}|\mathbf{x}_{j'})|$$

We note that this score is selecting the similar contexts from second task since it normalizes the attention score over the contexts of the second task. Symmetrically, we can define the attention score of context (i', j') from task 2 over (i, j) from task 1 as:

$$A'(j, i, j', i') = \frac{e^{\alpha'(j,i,j',i')}}{\sum_{(\hat{j},\hat{i}) \in L(j',i')} e^{\alpha(\hat{j},\hat{i},j',i')}}$$
$$\alpha'(j, i, j', i') = u(i, j)^T \text{diag}(\mathbf{a}') u(i', j')$$

Hence the context coupling error in reverse direction is given by:

$$CCE'(L, \mathbf{a}') = \sum_{((j,i),(j',i')) \in L} A'(j, i, j', i') \times |(s_i(y_{ji}|\mathbf{x}_j) - s'_{j'}(y'_{j',i'}|\mathbf{x}_{j'})|$$

Label coupling error In addition to the context coupling error defined above, we also take into account the explicit similarity between only labels,

using a character k-gram based embedding of the labels in the context (i, j) as: $v_{i,j}$. Hence, the label coupling error is given as:

$$LCE((i,j),(i',j')) = \text{SoftMax}(v_{i,j} \cdot v'_{i',j'}) \times |(s_i(y_{ji}|\mathbf{x}_j) - s'_{j'}(y'_{j',i'}|\mathbf{x}_{j'})|$$

LCE' is defined analogously. The label embeddings, $v_{i,j}$, are learned jointly with the model. The *total coupling error* between contexts (i, j) and (i', j') from the two tasks respectively, is the sum of context coupling error and the label coupling error:

$$TCE(L, \mathbf{a}, \mathbf{a}', \mathbf{v}) = \sum_{((i,j),(i',j'))\in L} [CCE((i,j),(i',j')) + CCE'((i,j),(i',j')) + LCE((i,j),(i',j')) + LCE'((i,j),(i',j'))]$$

We optimize the sum total of all the CRF losses and total coupling error in order to obtain model parameters. We use stochatic gradient descent, where minibatches are constructed from three lists: $\mathcal{D}^1$, $\mathcal{D}^2$, and L. Samples from the first two lists are used to calculate the CRF losses, while samples from L are used to calculate total coupling error, and the corresponding updates.

4 Experimental Results

In this section, we report results from our proposed method for multi-task attribute extraction, against single task attribute extraction. We implemented our model using tensorflow on a 8-core Centos machine. We used 300 dimensional pre-trained Glove vectors. We have also experimented with other customized word embeddings e.g. fasttext, but did not achieve significantly better results. For this work, we use single layer BiLSTM as the embedding layer. The hidden layer size for BiLSTM layer was set to 700. We have experimented with other embedding layer architectures, e.g. hidden layer sizes ranging from 300 to 900, and also two layer BiLSTMs with hidden layer sizes (500,700). However, the performance of single layer LSTM with hidden layer size 700 was found to be similar or better than others. For training, the batch size was chosen to be 30 for both the CRF loss batches and for coupling loss batches sampled from the shortlist L. ADAM was used as optimizer and we trained for maximum of 30 epochs. We trained the model for 30 epochs.

Table 1: Dataset Characterstics.

Vertical	# labels	# Examples (Train, Test)	# Exmpl. / label (max , min)
Jean	37	2206 , 948	1387,1
Trouser	38	1993, 856	1350, 1
Dress	30	4088, 1753	2241, 1
Mangalsutra	38	363, 157	333, 1
Chain	76	2068, 888	1195, 1
Jewellery	68	4863, 2085	4518, 1

Table 2: Similar Task Pairs for MTL

(Dress, Jean), (Mangalsutra, Jewellary)
(Trouser , Jean), (Chain, Jewellary),
(Mangalsutra, Chain)

Evaluation Metric As reported below, the datasets for this problem show extreme skew in terms of occurrence of labels. Hence, we use the standard metrics of *macro precision*, *macro recall*, and *macro F1 score*. We also report the micro-accuracy. While computing the macro-metrics (precision, recall and F1), we ignore the 'O' label. It is clear that macro-F1 score without the 'O' label, is the most representative metric here, from an application point of view.

4.1 Datasets

The dataset used here are taken from actual systems for product delivery used in Flipkart. We performed our experiments using data (both product descriptions, and ground truth annotations) from six verticals: `Dress`, `Jean`, `Mangalsutra`[1], `Chain`, `Trouser` and `Jewellery` available on Flipkart. These verticals are chosen based on three factors (1) GMV (Gross Merchandise Value), (2) Volume of data available and (3) Verticals with rich product descriptions. Number of labels in each vertical and number of tagged description in train and test data for each vertical is shown in table 1. The words in product descriptions for each vertical are tagged using `B,I,O` (short for beginning, inside, and outside) format where the B prefix before a tag indicates that the token is the beginning of a tag, and an I prefix before a tag indicates that the token is inside a tag and An O tag indicates that a token belongs to no tag.

Table 2 shows the pairs of similar tasks (verticals) which were trained togather for MTL. The pairs were chosen manually based on probability

[1] A type of Necklace

of occurrence similar labels in these tasks. The results for the each of the verticals is the best achieved for these pairs of tasks. Note that, while we have to manually provide a similar pair of tasks, the similarity between labels is automatically deciphered.

4.2 Performance Comparison

Table 3: Comparison of macro-F1 scores between single-task and multi-task models for various verticals

Vertical	Prec.	Recall	Acc.	F1
Single-task				
Dress	91.28	84.85	98.44	87.95
Jean	80.18	74.78	97.49	77.39
Mangalstr.	87.41	83.58	98.27	85.45
Trouser	78.52	71.91	98.83	75.07
Jewellery	71.06	72.13	97.94	**71.59**
Chain	58.61	49.63	96.59	53.75
Multi-task				
Dress	91.14	85.70	98.48	**88.23**
Jean	84.66	76.41	98.94	**80.32**
Mangalstr.	90.06	84.71	98.45	**87.30**
Trouser	78.94	75.32	98.90	**77.08**
Jewellery	70.94	69.60	97.83	70.26
Chain	64.51	55.90	96.75	**59.90**

In this section, we illustrate the effectiveness of our multi-task learning method. Table 3, reports the best performances of single and multi-task models for all the six verticals studied here. We can see that except for jewellery, multi-task model improve performance in terms of F1 score for all other verticals. For some verticals, e.g. chain, the improvement is more than 5 percent, while for other verticals the improvement lies in the 2 percent range. We note that the improvement depends on two main factors: whether we can find a close enough vertical to borrow from, and the number of examples already present in the current vertical. For example we can see that the vertical "Jewellery" has about 5000 examples, and also does not have a very close other vertical to borrow information from. Hence in it's case MTL is not able to improve the performance.

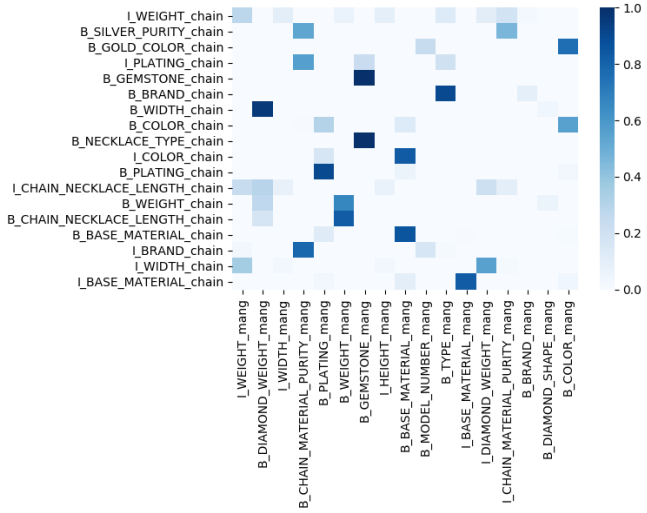
In table 4, we report the fine-grained improvements of top 5 labels for the verticals: Trouser, Jean, Mangalsutra, and Chain. We note that the top improvements for these verticals are in the range of 51%, 46%, 29% and 22% respectively. We also note that number of examples for these labels in the training dataset (#ex column) are respectively 6, 15, 6, and 7. Hence this table further corroborates our claim that MTL improves the performance for labels with lower amount of information in the single task training set.

Table 4: Attribute-wise percentage improvement on various tasks

Attribute	#Ex.	Task	Prec.	Recall	F1-Sc.	%Imp.
Trouser						
I_occas.	6	single multi	1.0 1.0	0.17 0.67	0.29 0.8	0.51
I_suitab.	3	single multi	1.0 1.0	0.33 0.67	0.5 0.8	0.3
B_suitab.	6	single multi	0.4 0.43	0.33 0.5	0.36 0.46	0.1
I_brand	283	single multi	0.91 0.92	0.88 0.92	0.89 0.92	0.03
B_pattern	308	single multi	0.88 0.88	0.9 0.95	0.89 0.92	0.03
Jean						
I_pattern	15	single multi	0.33 1.0	0.07 0.4	0.11 0.57	0.46
I_suitab.	2	single multi	0.5 1.0	0.5 0.5	0.5 0.67	0.17
B_suitab.	5	single multi	0.5 1.0	0.4 0.4	0.44 0.57	0.13
I_ref._fit	14	single multi	1.0 1.0	0.57 0.64	0.73 0.78	0.05
B_pattern	206	single multi	0.9 0.92	0.89 0.91	0.89 0.91	0.02
Mangalsutra						
I_diamnd	6	single multi	0.0 1.0	0.0 0.17	0.0 0.29	0.29
B_diamnd	19	single multi	0.82 0.85	0.47 0.58	0.6 0.69	0.09
B_chain	19	single multi	0.94 0.94	0.84 0.89	0.89 0.92	0.03
B_brand	112	single multi	0.97 0.97	0.92 0.97	0.94 0.97	0.03
I_gemst.	37	single multi	1.0 1.0	0.89 0.92	0.94 0.96	0.02
Chain						
I_warr.	7	single multi	0.64 1.0	1.0 1.0	0.78 1.0	0.22
I_weight	51	single multi	0.68 0.76	0.8 0.88	0.74 0.82	0.08
I_width	18	single multi	0.68 0.75	0.83 0.83	0.75 0.79	0.04
B_weight	26	single multi	0.76 0.79	0.85 0.88	0.8 0.84	0.04
I_color	92	single multi	0.93 0.86	0.59 0.67	0.72 0.76	0.04

4.3 Validation of Attribute Similarity

In this section, we validate the learned attribute-attribute similarity, by studying the attribute-wise F1-scores for the similar attribute pairs. Figure 4-(a) shows the full attention heatmap for all labels between the pair of tasks: Mangasutra - Chain. Here the attention is normalised over the attributes of y-axis (task chain). It is clear from the heatmap



(a)

Borrower-Chain	Lender-Mangalstr.(Att.[2])
B_base_mat.	B_base_mat. (0.86)
I_weight	I_weight (0.28)
B_necklace_type	B_gemstone (1.0)
B_color	B_color (0.56)
B_gemstone	B_gemstone (1.0)
I_brand	B_mat._purity (0.79)
B_plating	B_plating (0.9)
B_brand	B_type (0.9)
I_base_mat.	I_base_mat. (0.83)
B_weight	B_weight (0.67)

(b)

Figure 4: (a) Attention Heatmap between labels of Chain (task 1) and Mangalsutra (task 2). (b) Attribute pairs with highest attention values, and the corresponding attention score.

that attention mechanism is indeed choosing the similar labels between the pairs of tasks, irrespective of whether there is an improvement in accuracy for the pair of labels.

In figure 4-(b), we report the topmost pairs of labels with the highest attention scores, along with the corresponding increase in accuracy. The left column borrower labels(Chain) and right column shows Lender labels(Mangalsutra) which got the highest average attention weights across all contexts-pairs in the list L. The value in brackets shows the attention value. The bold entries appear in top-5 attributes, with highest F1-scores in table 4. One can also see non-obvious correspondences, e.g. `Necklace_type` from chain can borrow all the information from `Gemstone` from lender vertical Mangalsutra. We can also see that in most of the cases, the labels from task 1 borrow information from corresponding labels of task 2, even though this information was not explicitly furnished. This observation provides us further confidence that the attention mechanism used for discovery of similar labels and similar contexts, indeed works effectively.

This observation further validates the effectiveness of our attention model in extracting similar pairs of labels between two tasks using the coupling loss. We believe this mechanism can be applied in many more situations to shortlist important and similar attributes in other contexts, while jointly learning a prediction model.

5 Conclusion

In this paper, we study attribute-value extraction from production description in the e-commerce domain. Many of the attributes occur in very few descriptions. Hence the amount of supervised training data available for these attributes is very low, which leads to low prediction performance We thus propose a novel multi-task learning based algorithm which borrows information from related domains (i.e., category/vertical) in order to improve prediction performance of infrequently occurring attributes. We validate the proposed method with extensive experimental evaluation on a large dataset of six verticals from a prominent, e-commerce company. The proposed technique not only achieves higher accuracy on verticals with similar labels, but also can be used for discovering attribute similarities across verticals.

References

Omar Alonso, Vasileios Kandylas, and Rukmini Iyer. 2019. Unsupervised Construction of a Product Knowledge Graph. In *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42st International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019 (CEUR Workshop Proceedings)*, Jon Degenhardt, Surya Kallumadi, Utkarsh Porwal, and Andrew Trotman (Eds.), Vol. 2410. CEUR-WS.org. http://ceur-ws.org/Vol-2410/paper23.pdf

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML '08*.

Varun Embar, Bunyamin Sisman, Hao Wei, Xin Luna Dong, Christos Faloutsos, and Lise Getoor. 2020. Contrastive Entity Linkage: Mining Variational Attributes from Large Catalogs for Entity Linkage. In *Automated Knowledge Base Construction*. https://openreview.net/forum?id=fR44nF03Rb

Mahesh Joshi, Ethan Hart, Mirko Vogel, and Jean-David Ruvini. 2015. Distributed Word Representations Improve NER for e-Commerce. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Association for Computational Linguistics, Denver, Colorado, 160–167. https://doi.org/10.3115/v1/W15-1522

Giannis Karamanolakis, Jun Ma, and Xin Dong. 2020. TXtract: Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories. *ArXiv* abs/2004.13852 (2020).

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)*. Association for Computational Linguistics, Berlin, Germany, 107–111. https://doi.org/10.18653/v1/P16-2018

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*. Association for Computational Linguistics, San Diego, California, 260–270. https://doi.org/10.18653/v1/N16-1030

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The Benefit of Multitask Representation Learning. *Journal of Machine Learning Research* 17, 81 (2016), 1–32. http://jmlr.org/papers/v17/15-242.html

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

Martín Rezk, Laura Alonso Alemany, Lasguido Nio, and Ted Zhang. 2019. Accurate Product Attribute Extraction on the Field. *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (2019), 1862–1873.

Sourangshu Bhattacharya Anirban Dasgupta Soumen Chakrabarti Rijula Kar, Susmija Reddy. 2018. Task-Specific Representation Learning for Web-scale Entity Disambiguation. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *ArXiv* abs/1706.05098 (2017).

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 47–55.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up Open Tagging from Tens to Thousands: Comprehension Empowered Attribute Value Extraction from Product Title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5214–5223. https://doi.org/10.18653/v1/P19-1514

Yongxin Yang and Timothy M. Hospedales. 2017. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. *ArXiv* abs/1605.06391 (2017).

Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. OpenTag: Open Attribute Value Extraction from Product Profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1049–1058. https://doi.org/10.1145/3219819.3219839

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal Joint Attribute Prediction and Value Extraction for E-commerce Product. *arXiv preprint arXiv:2009.07162* (2020).