

AtTGen: Attribute Tree Generation for Real-World Attribute Joint Extraction

Yanzeng Li^{1,2}, Bingcong Xue¹, Ruoyu Zhang¹, Lei Zou^{1,3*}

¹Wangxuan Institute of Computer Technology, Peking University, Beijing, China

²National Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China

³TopGraph.AI

liyanzeng@stu.pku.edu.cn

{xuebingcong, ry_zhang, zoulei}@pku.edu.cn

Abstract

Attribute extraction aims to identify attribute names and the corresponding values from descriptive texts, which is the foundation for extensive downstream applications such as knowledge graph construction, search engines, and e-Commerce. In previous studies, attribute extraction is generally treated as a classification problem for predicting attribute types or a sequence tagging problem for labeling attribute values, where two paradigms, i.e., closed-world and open-world assumption, are involved. However, both of these paradigms have limitations in terms of real-world applications. And prior studies attempting to integrate these paradigms through ensemble, pipeline, and co-training models, still face challenges like cascading errors, high computational overhead, and difficulty in training. To address these existing problems, this paper presents *Attribute Tree*, a unified formulation for real-world attribute extraction application, where closed-world, open-world, and semi-open attribute extraction tasks are modeled uniformly. Then a text-to-tree generation model, *AtTGen*, is proposed to learn annotations from different scenarios efficiently and consistently. Experiments demonstrate that our proposed paradigm well covers various scenarios for real-world applications, and the model achieves state-of-the-art, outperforming existing methods by a large margin on three datasets. Our code, pre-trained model, and datasets are available at <https://github.com/lsvih/AtTGen>.

1 Introduction

Attribute Extraction (AE) is a practical application of the Information Extraction (IE) task, aiming to identify the attribute name and the corresponding attribute value from unstructured or semi-structured text fragments (Ghani et al., 2006; Ravi and Pasca, 2008; More, 2016). Figure 1 shows a typical product profile with extracted attribute tags.

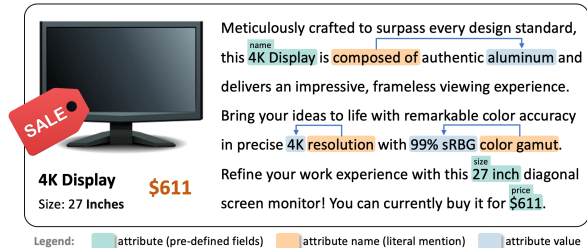


Figure 1: An example of attribute extraction, highlighted with annotations in different tagging forms.

As the foundation for various downstream applications such as knowledge graph construction, search engines, e-Commerce and recommender systems, AE has attracted extensive research interest in recent years (Zheng et al., 2018; Xu et al., 2019; Zhu et al., 2020; Jain et al., 2021; Zhang et al., 2022; Li and Zou, 2022).

There are two basic subtasks in the research of AE, namely, attribute name extraction and attribute value extraction. And we use the RDF-style triple¹ $\langle e, n, v \rangle$ to denote the entity, attribute name, and attribute value respectively. According to whether the attribute name set is pre-defined, AE can be divided into two paradigms, i.e., the *Closed-World Assumption (CWA)* and the *Open-World Assumption (OWA)*. For CWA AE, the attribute name n is limited to a finite set of the pre-defined schema, where attribute name extraction is typically modeled as a classification task (Zeng et al., 2014; Zhou et al., 2016), and attribute value extraction models are trained for each target attribute (Zheng et al., 2018; Zhu et al., 2020; Yan et al., 2021). While for OWA AE, which is also known as “New Attribute Discover” (Wong and Lam, 2010; Zhang et al., 2022) and “Open Information Extraction” (Cui et al., 2018), the attribute name is schema-free and can be extracted from the text. Sequence tagging methods are broadly employed to extract those attributes (Xu et al., 2019). Recently, researchers

*Corresponding Author

¹<https://www.w3.org/TR/n-triples/>

also explore novel paradigms such as Question Answering (QA) models (Wang et al., 2020; Shinzato et al., 2022; Yang et al., 2022) and generative models (Roy et al., 2022) to generalize the ability of attribute extraction.

However, AE in the real world is far more complicated. On the one hand, in closely related fields like e-commerce, new types of products with new sets of attributes are so constantly arising that the pre-defined schema is never enough. For example, an analysis in Zhang et al. (2022) has shown that only 30 / 51 attributes are found in existing structured product profiles of Amazon’s 10 product types. On the other hand, however, attribute extraction methods shouldn’t overlook the huge value and commonalities behind known attributes, and it is inherent that not all attributes can be fully identified by open extraction methods due to the lack of literal name mentions, e.g. name and size in Figure 1. It is possible to carry out both CWA and OWA methods when needed, just as Zhang et al. (2021) attempts preliminarily. But apart from the fragmentation of the problem form and the unnecessary computing overhead, a more prominent issue is that such simple integration neglects the natural connections between the CWA vocabulary and the OWA ability in attribute extraction, and thus cannot achieve satisfactory results. In this paper, we, for the first time, explicitly unify the different AE paradigms in the form of *Attribute Tree*, and present a text-to-tree based generative model called AtTGen to solve the real-world attribute joint extraction task.

Specifically, our proposed AtTGen successfully implements the unification of attribute tagging and classification tasks by generating the *Attribute Tree*, and congenitally circumvents the problem of “null”-value that troubles pioneers (Xu et al., 2019; Wang et al., 2020). Further, the head entity is optional as the root node on *Attribute Tree* to meet the actual situation, as well as to enhance the extraction performance with the help of the subject guidance (Yu et al., 2021; Zhang et al., 2021). AtTGen reduces the length of the generated sequence and thus shrinks the search space by conducting the tree generation model. And it can accurately mark out the span of attribute values and extract unseen attributes with the pointer-copy mechanism (Zhou et al., 2018). Moreover, the *teacher forcing manner* (Williams and Zipser, 1989) and the converted path-generation training objective further reduce the exposure bias (Zhang et al., 2020) to improve

the generalization and effectiveness.

In short, the major contributions of this paper can be summarized as follows:

- We are the first to define different attribute extraction paradigms like CWA, OWA and semi-open as the attribute tree generation problem, formally unifying multiple tasks and fully capturing the internal connections.
- We design a novel text-to-attribute tree generation model with a pointer-based copy mechanism for extracting both literal mentions and category labels.
- We evaluate our model on several benchmark datasets. Experimental results show that our method achieves state-of-the-art (SOTA) and outperforms existing works by a large margin in all scenarios including open, semi-open and closed-world attribute extraction.

2 Preliminary

We first formalize the definition of two mainstream paradigms widely used in Attribute Extraction.

Definition 1 (Closed-World Assumption). CWA AE receives a descriptive text $\mathcal{T} = [t_1, t_2, \dots]$, e.g. a product title, and a pre-defined schema \mathcal{A} which contains a set of attributes (i.e., attribute vocabulary) to extract all attribute pairs $\langle n, v \rangle$ for a possibly given head entity e , where $n \in \mathcal{A}$ is the attribute name (also called attribute type), and $v \in \mathcal{T}$ is the attribute value extracted from the text.

Definition 2 (Open-World Assumption). OWA AE takes a descriptive text $\mathcal{T} = [t_1, t_2, \dots]$ as input, and the target is to discover all attribute pairs $\langle n, v \rangle$ for a possibly given head entity e , where both the attribute name n and the attribute value v are from the given text, i.e. $n \in \mathcal{T}$ and $v \in \mathcal{T}$.

As stated in Section 1, individual one of the above paradigms does not always work well in real-world applications, and the pipeline approach adopted by Zhang et al. (2021) to merge the results of the two paradigms would introduce problems such as cascading errors. Therefore, we propose a formal definition of real-world AE and its solution in the following sections.

3 Problem Formalization

Section 1 has expounded that attribute extraction in real-world applications sometimes needs both the

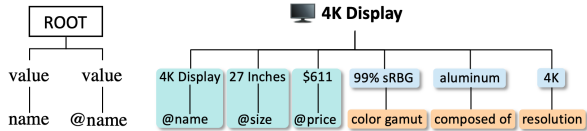


Figure 2: The abstract illustration of *Attribute Tree* (left) and an instantiated one describing the attributes of the example in Figure 1 (right). The attribute names starting with “@” represent those stemming from the schema.

guidance of the schema and the ability to extract free attributes from texts. It is actually an extensive aggregation covering both CWA and OWA AE, as well as a semi-open scenario where attribute names can be obtained from both. Therefore we formally define the real-world attribute extraction as:

Definition 3 (Real-world Attribute Extraction). Given a text \mathcal{T} , and an optional \mathcal{A} , “real-world AE” is to fill the explicit slots for the optional category in \mathcal{A} , or to dig more free attributes from \mathcal{T} , or to capture attributes from both \mathcal{A} and \mathcal{T} . i.e., the final result of real-world AE is a set of attribute pairs $\langle n, v \rangle$ where $v \in \mathcal{T}$, $n \in \mathcal{H} = \{\mathcal{A}, \emptyset\} \cup \{\mathcal{T}, \emptyset\}$ and $\mathcal{H} \neq \emptyset$.

To implement such an extraction paradigm uniformly, we devise a principled structure, *Attribute Tree*, to formally model the target of all real-world AE circumstances:

Definition 4 (Attribute Tree). An attribute tree T for a descriptive sentence $sent$ is an unweighted tree with a fixed height $h = 2$. All the branches of the tree T have a determined order (r, v, n) , and the root r is the only entry node that can be either empty \emptyset or the head entity (also called the subject) $subj$ of the attributes.

Figure 2 visualizes the attribute tree and its instances. The path from the root to the leaves is also the reasoning path of the proposed model. Borrowing the notation from epistemology (Martin-Löf, 1996), there are:

$$\begin{aligned} \{sent, r\} \vdash v \\ \{sent, r, v\} \vdash n \quad r \in \{\emptyset, subj\} \end{aligned} \quad (1)$$

which means the attribute value v is derived from the original sentence $sent$ and the root node r ; and the attribute name n , whether coming from the input text or the given schema, can be predicted by the integrated information from the sentence, the attribute value, and the root node. This kind of path order can naturally evade the insignificant

“NULL” value problem pointed out by Shinzato et al. (2022).

Definition 5 (Subject Guidance). Setting the subject $subj$ of a descriptive sentence $sent$ as the root node r of the corresponding attribute tree T when available, i.e. let $r = subj$ in Equation 1, is called enabling the *subject guidance*.

As attributes typically characterize entities and are strongly bound to the subject, we naturally introduce the subject guidance for AE in such a way and the effectiveness has been preliminarily demonstrated in Yu et al. (2021); Zhang et al. (2021).

4 Methodology

We design a unified tree generative model AtTGen, committing to jointly extracting attribute names and values under various scenarios in the real world. It is partially inspired by the success of Seq2Tree models (Dong and Lapata, 2016; Liu et al., 2019; Zhang et al., 2020) and pointer-copy based span-selector (Zhou et al., 2018; Ma et al., 2022) in other tasks. The overall architecture is shown in Figure 3, and we demonstrate the model details in the following subsections.

4.1 Encoder

We employ the classical BiLSTM-CNN (Chiu and Nichols, 2016) neural network to encode the input text into a continuous latent space². Given a sequence input $[t_1, t_2, \dots, t_n]$, the encoded text representation $\mathbf{h}_t \in \mathbb{R}^{m \times n}$ is obtained by:

$$\begin{aligned} \mathbf{h}_t &= \text{Encoder}(sent) \\ &= \text{Conv}_{\text{enc}}(\text{BiLSTM}_{\text{enc}}(\text{Emb}(sent))) \end{aligned} \quad (2)$$

in which Emb is to gain the embedded vector of tokens from the lookup table and m is the dimension of the embedding, $\text{BiLSTM}_{\text{enc}}$ is Bidirectional Long Short-Term Memory network (Hochreiter and Schmidhuber, 1997) for modeling the dependencies of the input sequence, and Conv_{enc} is Convolutional Network (Collobert et al., 2011) for extracting features from the encoded text representation. Meanwhile, the category labels of attribute names from the given schema also contain useful semantic information for generating the attribute tree, thus we use the same encoder to obtain the label representation of the attribute names as:

$$\mathbf{h}_l = \text{Encoder}(labels) \quad (3)$$

²Adapting PLMs to AtTGen is discussed in Section 8

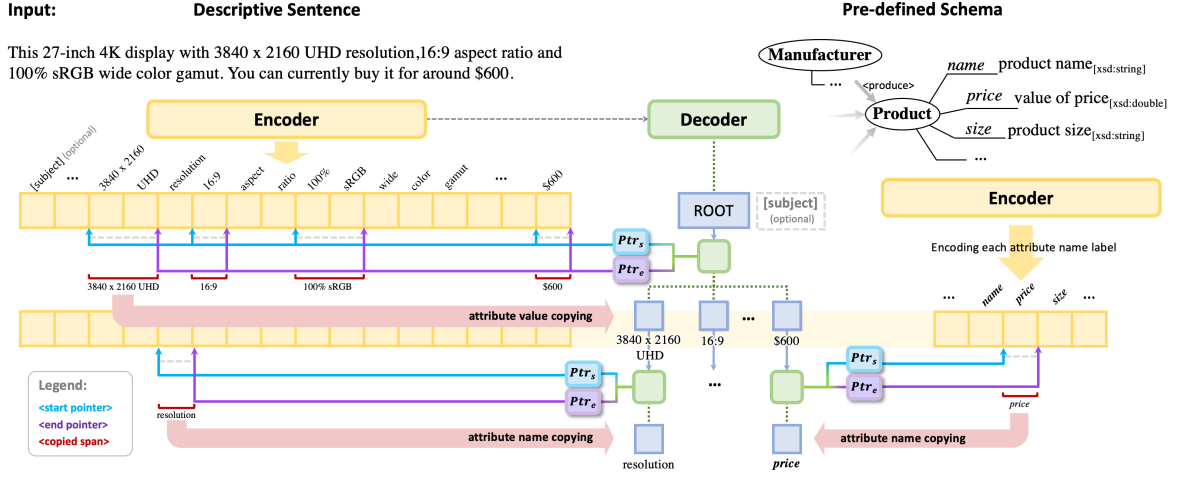


Figure 3: The overview of AtTGen (Best viewed in color). The blocks in yellow, green, and blue, denote the encoded text representation, the tree decoder, and the obtained attribute tree respectively, and the red arrows represent the direction of copying.

Then we can concatenate the two parts and get the initial root node representation as $h_r = \text{Encoder}([sent||labels])$, which allows the successor decoders to uniformly generate nodes from both the input sentence and the category label set.

In addition, the subject of the attribute would be concatenated with the input sentence as $[\langle subject \rangle, [sep], t_1, \dots, t_n]$ for the subject guidance, in which $[sep]$ is a separator token.

4.2 Tree Decoder

The decoding target of our method is to generate a structured attribute tree. As a tree can be divided into several paths from the root node to the leaf node, the generation of a tree can also be decomposed into the problem of generating multiple paths. Therefore, the decoder of AtTGen is denoted as:

$$rs, \mathbf{h}_{rs}, \mathbf{s}_t = \text{Decoder}(\mathbb{T}, \mathbf{h}_p, \mathbf{s}_{t-1}) \quad (4)$$

where rs is the generated result, \mathbf{h}_{rs} is the representation of the decoded tokens, \mathbf{s}_t and \mathbf{s}_{t-1} are the current and the previous state of the decoder respectively. Each decoding step relies on several inputs: (1) the target space of decoding \mathbb{T} , which is to limit the selection range of the final result of the decoder and thus shrinks the search space; (2) the representation of the antecedent path \mathbf{h}_p ; (3) the state of the decoder \mathbf{s}_t , used to determine the currently decoded node is at what level of the attribute tree.

Specifically, given the input \mathbf{h}_p and the previous decoding state \mathbf{s}_{t-1} , a unary LSTM is employed for decoding the state \mathbf{s}_t as:

$$\mathbf{s}_t = \text{LSTM}_{\text{dec}}(\mathbf{h}_p, \mathbf{s}_{t-1}) \quad (5)$$

The decoding feature \mathbf{h}_{rs} for generating results is obtained by a convolutional network Conv_{dec} with an attention-based weighted sum like (Bahdanau et al., 2015) as:

$$h_{rs} = \text{Conv}_{\text{dec}}(\text{Att}(\mathbf{h}_t, \mathbf{s}_t)) \quad (6)$$

Then the final result as follows is decoded from the pointer-based *span copier* (Ptr) explained in Section 4.3:

$$\begin{aligned} \mathbf{i}_{\text{start}}, \mathbf{i}_{\text{end}} &= Ptr_s(\mathbf{h}_{rs}), Ptr_e(\mathbf{h}_{rs}) \\ rs &= \mathbb{T}[\mathbf{i}_{\text{start}} : \mathbf{i}_{\text{end}}] \end{aligned} \quad (7)$$

The whole decoding process for AtTGen is described in Algorithm 1.

Algorithm 1: Attribute Tree Decoder

Input : A descriptive sentence: $sent$
A category set from flattened schema: $labels$

Output : The attribute tree of $sent$

```

// Decoding attributes from plain text and
pre-defined schema jointly.
1  $h_r \leftarrow \text{Encoder}([sent||labels])$ 
2 if use subject guidance then
3    $r, h_r, s_r \leftarrow \text{Decoder}(sent, h_r, \emptyset)$ 
4    $root \leftarrow \text{Tree}(r)$ 
5 else
6    $s_r \leftarrow \emptyset$ 
7    $root \leftarrow \text{Tree}(placeholder)$ 
8  $v, h_v, s_v \leftarrow \text{Decoder}(sent, h_r, s_r)$ 
9 for  $v, h_v$  in  $v, h_v$  do
10   $h_v = h_r \oplus h_v$ 
11   $n, h_n, s_n \leftarrow \text{Decoder}([sent||labels], h_v, s_v)$ 
12  for  $n, h_n$  in  $n, h_n$  do
13    if  $v \notin root.children()$  then
14       $root.add\_child(v)$ 
15     $root.find\_child(v).add\_child(n)$ 
16 return  $root$ 

```

where \emptyset is a randomly initialized vector to represent the initial decoding state. r , h_r and s_r are the

decoder’s output for the root node (the optional subject), representing the generated result, the hidden representation and the current state respectively. Similarly, $(\mathbf{v}, \mathbf{h}_v, s_v)$ and $(\mathbf{n}, \mathbf{h}_n, s_n)$ are the other two sets of outputs from the decoder, for the decoding process of attribute values and attribute names respectively. Note that if subject guidance is enabled, the decoder will update \mathbf{h}_r by decoding subject firstly, and construct the root node of the tree (Line 2-4), otherwise the root node is replaced by a placeholder (Line 5-7). The attribute values and attribute names are sequentially decoded in the order of Equation 1 to construct *Attribute Tree* as shown in Line 8-15 in Algorithm 1.

4.3 Span Copier

We propose to use a unified *span copier* to ensure the spans are correctly copied from the original sentence or the label set during the decoding process.

$$\begin{aligned} Ptr_s(\mathbf{h}) &= \sigma(\mathbf{W}_s \mathbf{h} + \mathbf{b}_s) \\ Ptr_e(\mathbf{h}) &= \sigma(\mathbf{W}_e \mathbf{h} + \mathbf{b}_e) \end{aligned} \quad (8)$$

in which \mathbf{W}_s and \mathbf{W}_e are trainable weights, \mathbf{b}_s and \mathbf{b}_e are trainable bias, \mathbf{h} denotes the hidden state of the current decoding step, and σ is the sigmoid active function. The $Ptr_{(\cdot)}$ produces a constant vector that denotes the start/end index of the copied span. For those nodes in the closed-world setting whose mention does not exist in the original text (e.g., name, size, and price in Figure 1), we further add an equality constraint $Ptr_s = Ptr_e$, restricting the pointers to select only one category label when decoding from the label set, which reduces generative errors and improves the training efficiency.

4.4 Training Objective

In the decoding process, we apply *teacher forcing manner* (Williams and Zipser, 1989) for efficient training and encourage the model to reduce the distance of all paths between the generated tree and the ground truth:

$$\begin{aligned} L_{path} &= \delta \sum_{i \in \{s,e\}} \text{BCE}(Ptr_i(\mathbf{h}_r), y_{i_r}^*) \\ &+ \sum_{j \in \{v,n\}} \sum_{i \in \{s,e\}} \text{BCE}(Ptr_i(\mathbf{h}_j), y_{i_j}^*) \end{aligned}$$

where $\delta \in \{0, 1\}$ indicates whether to enable the subject guidance; $y_{s_{(\cdot)}}^*/y_{e_{(\cdot)}}^*$ denotes the golden standard start/end index of either a literal mention

or a category label of the target span; $\mathbf{h}_{(\cdot)}$ represents the hidden state of the decoder to distinguish the level it is decoding. BCE is the Binary Cross Entropy loss to optimize the prediction of the index vectors individually for each step:

$$\text{BCE}(y, y^*) = -\frac{1}{N} \sum_{i=1}^N y_i^* \cdot \ln y_i + (1 - y_i^*) \cdot \ln(1 - y_i)$$

where N is the length of the input sentence, y_i is the predicted probability of the i -th element and y_i^* is the corresponding ground truth.

5 Experiments

5.1 Experimental Setup

Datasets. We conduct our experiments on three publicly available datasets to examine the capacity and the generality of our model over various real-world AE settings:

MEPAVE (Close-World Benchmark)³ (Zhu et al., 2020) is a multimodal e-Commerce product attribute extraction dataset, which contains 87k product description texts (in Chinese) and images, involving 26 types of attributes. We follow the same dataset settings as Zhu et al. (2020), except that we leave the visual information and use the description texts only.

AE-110K (Open-World Benchmark)⁴ (Xu et al., 2019) is a collection of 110k product triples (in English) from AliExpress with 2,761 unique attributes. It can well measure the open extraction ability and generation performance of different models. We split this dataset via the cleaning script of Shinzato et al. (2022), and remove invalid and “NULL” value attributes following Roy et al. (2022).

Re-CNShipNet (Semi-Open Benchmark) is a revised version of the functional attribute extraction dataset CNShipNet⁵ (Zhang et al., 2021), where numerical attributes account for the majority to bring new challenges. We manually fix the incorrect annotations in the old version and rebalance the ratio of closed- to open-setting labels (Li et al., 2021). Now it contains about 5k entity-attribute instances (mostly in Chinese), among which 40% obtain attributes from the literal texts and others are within 9 pre-defined attribute types.

Baselines. We compare the proposed model with several strong and typical baselines including:

³<https://github.com/jd-aig/JAVE>

⁴https://github.com/lanmanok/ACL19_Scaling_Up_Tagging/blob/master/publish_data.txt

⁵<https://github.com/lsvih/SOAE>

1) Sequence Tagging-based methods, a kind commonly adopted in IE which typically uses semantic tags such as BIO to identify the extracted items: **RNN-LSTM** (Hakkani-Tür et al., 2016), **Attn-BiRNN** (Liu and Lane, 2016), and **BiLSTM-CRF** (Huang et al., 2015) are all specially designed RNN-based models for modeling the intent of classification and extraction tasks. **ScalingUp** (Xu et al., 2019) is a BERT-based model to extract attribute values with BiLSTM to perform interaction attention between attribute names and values.

2) PLM-based methods: **BERT** (Devlin et al., 2019) is a well-known pre-trained language model (PLM) and we follow the vanilla setting of classification and sequence tagging tasks, **Joint-BERT** (Chen et al., 2019) is a variant of BERT to solve slot filling and classification jointly.

3) Joint IE-based (JE) methods, which originate from the entity-relation extraction task and typically extract entities and classify relations in a cascading fashion: **ETL-Span** (Yu et al., 2020) and **CasRel** (Wei et al., 2020) are two classic JE models for relation extraction and we adapt them to the AE task here. **SOAE** (Zhang et al., 2021) achieved SOTA on CNShipNet by merging the results of a JE model and a classification model. **JAVE** (Zhu et al., 2020) is an attention-based attribute joint extraction model and **M-JAVE** further takes advantage of multimodal information, and they were the best models for MEPAVE.

4) Sequence Generative Model: We also implement the latest word sequence generation method (Roy et al., 2022) based on the large-scale pre-trained **BART** (Lewis et al., 2020) model.

We conduct the baselines and adapt them to the target datasets accordingly. See Appendix A for implementation details.

Metrics. Following previous works (Zheng et al., 2018; Xu et al., 2019; Zhu et al., 2020; Zhang et al., 2021), we use F1 score as the metric and adopt *Exact Match* criteria (Wei et al., 2020), in which only the full match to the ground truth is considered correct. We report the results of attribute name and value extraction respectively as Zhu et al. (2020).

5.2 Main Results

This section presents the overall results of the models over various AE scenarios in Table 1, 2, and 3. In general, we can observe that our model outperforms the baselines over all three scenarios in real-world AE.

Model	Attribute	Value
RNN-LSTM	85.76	82.92
Attn-BiRNN	86.10	83.28
BERT	86.34	83.12
Joint-BERT	86.93	83.73
ScalingUp (BERT-based)	-	77.12
CasRel (BERT-based)	84.74	79.61
JAVE (LSTM based) [‡]	87.88	84.09
JAVE (BERT based) [‡]	87.98	84.78
M-JAVE (LSTM-based) ^{†‡}	90.19	86.41
M-JAVE (BERT-based) ^{†‡}	90.69	87.17
AtTGEN (LSTM-based, Ours)	96.48	96.26

Table 1: Experimental results on MEPAVE (CWA). † denotes the method utilizing image information. ‡ represents the result is from the original paper.

Model	Attribute	Value
RNN-LSTM	36.79	20.86
BiLSTM-CRF	40.25	37.51
ScalingUp (BERT-based)	-	31.67
BERT	54.01	52.42
CasRel (BERT-based)	56.92	53.73
JAVE (BERT-based)	53.82	38.25
BART (Seq. Gen.)	58.46	53.32
AtTGEN (LSTM-based, Ours)	57.60	59.77

Table 2: Experimental results on AE-110K (OWA).

As shown in Table 1, our model achieves a big improvement in the closed-world AE task. Even though the previous SOTA model (M-JAVE BERT version) introduces PLM and takes advantage of extra multimodal information (product images), we still gain a 9.09% improvement in attribute value extraction and 5.79% in attribute name prediction.

In the open setting shown in Table 2, AtTGen consistently performs well in attribute value extraction, with a 6.45% improvement than BART, an elaborate and dedicated PLM-based model. It has a slightly lower result compared with BART when extracting attribute names (0.86%), due to the absence of the semantic knowledge contained in the large-scale PLMs for efficiency issues. We will consider introducing such knowledge in future work, which we believe will further improve the performance. But the current results are still strong enough to demonstrate the open extraction capability of our model.

As for the semi-open scenario displayed in Table 3, our model again outperforms CasRel, a strong joint model in the information extraction field. We

Model	Attribute	Value
RNN-LSTM	53.6	52.9
Attn-BiRNN	51.9	52.0
BERT	58.3	57.8
Joint-BERT	59.1	58.4
ScalingUp (BERT-based)	-	56.1
ETL-Span	66.7	65.6
CasRel (LSTM-based)	66.5	67.2
CasRel (BERT-based)	70.1	69.7
SOAE (BERT-based)	69.4	69.0
AtTGEN (LSTM-based, Ours)	73.4	75.4

Table 3: Experimental results on Re-CNShipNet (Semi).

also attain better results than SOAE, which was the SOTA on this dataset by conducting both OWA and CWA models. This can be credited to our unified attribute tree model to naturally capture the intrinsic connections in the partial-closed world.

It can be concluded that, as the first to design a tree generative model in AE, our method can be silkily adapted to different real-world scenarios at a small cost, and achieves remarkable results whether the dataset is in the e-Commerce domain (MEPAVE, AE-110K) or news (Re-CNShipNet), and whether the language of the datasets is English (AE-110K) or Chinese (MEPAVE and Re-CNShipNet). Moreover, unlike quite many baselines relying on external knowledge in the large-scale language models, we achieve outstanding results by training from scratch, and thus has a dominant advantage in the parameter-efficiency (e.g., BERT has ~ 110 M parameters, BART has ~ 139 M, AtTGen has only ~ 2 M). We hypothesize that the superiority comes from the unified problem formalization as well as the novel tree generation model design. On the one hand, our model keeps the simplicity as a generation model, providing a unified way to capture the semantic associations between open and closed vocabulary, and between attribute names and values. On the other hand, different from traditional Seq2Seq models that decode all triples autoregressively into a linear sequence, our tree structure decomposes the decoding target into several paths of length three, removing the unnecessary order among different triplets and effectively alleviating the exposure bias problem in long-distance generation tasks (Zhang et al., 2020).

Furthermore, we notice that the performance of the models varies across different datasets, which can be attributed to the varying levels of complexity

Variant	MEPAVE	AE-110K	R-CSN
AtTGen	96.14	56.85	73.21
w/o subject guidance	-	-	70.06
w/o span copier	89.20	49.16	61.59
repl. (r, n, v) path order	95.12	49.39	67.58
w/o schema	-	-	42.73

Table 4: Ablation results measured by Exact Match F1 score of attribute pairs. “-” denotes the setting is not appropriate to the corresponding dataset; R-CSN is the abbreviation for Re-CNShipNet.

and quality of the datasets. For example, MEPAVE is a well-annotated benchmark with only a small number of attribute types, hopefully for better results. While AE-110K suffers an inevitable long-tail distribution problem, and Re-CNShipNet is limited by the data scale and the uncertain ratio of CWA/OWA labels, posing greater challenges and leading to the results that all models still have a large room for improvement.

5.3 Ablation Study

In this section, we carry out several ablation experiments to study the effectiveness of each sub-component in *AtTGen*. The whole results are listed in Table 4 and we can find these phenomenons:

1) The performance reduces by 3.15% on Re-CNShipNet dataset without the subject guidance, indicating **the usefulness to exploit the constraint semantics of the subject in attribute extraction**. Along with the findings in Yu et al. (2021); Zhang et al. (2021), we may conclude that subject guidance is a powerful enhancement in various information extraction situations.

2) We remove the span copier by replacing it with an ordinary token generator to extract values from the whole vocabulary. It can be seen that the performance drops by 8.75% on average, and the degradation is more evident in the open and semi-open settings, where the performances are down to the same level as other sequence tagging-based models. This proves that the advantage of the model largely comes from the copy mechanism to detect boundary information of the spans rather than directly modeling the attributes. We therefore say that **span copier can play a prominent role in AE**.

3) We also explore the influence of the generation order in *Attribute Tree* and the results show that changing the path order from (r, v, n) to (r, n, v) slightly reduces the effect (4.7% averagely). Somewhat different from a prior experiment conducted

in (Zhang et al., 2020), which shows that in entity-relation joint extraction task, relations should come first to get the best performance, our conclusion here is that **attribute values should be extracted before attribute names, especially in open scenarios**. One possible explanation for this difference between relation and attribute extraction is that attribute values typically have more evident patterns to trigger the following attribute name prediction. Besides, the path order of (r, v, n) is able to reduce the confusion of multifarious attribute names and well evades the “NULL” value problem.

4) Removing schema information directly deprives the model’s capacity to learn from the existing ontology, and significantly degrades its performance on the Re-CNShipNet dataset, showing that **pre-defined schema can strengthen models’ applicability in real-world AE applications**.

By these ablation studies, we have not only demonstrated that each delicate design in our model plays an important role, but proposed several interesting findings which we believe will shed some light for future research.

5.4 Case Study

We present two case studies from Re-CNShipNet dataset to further illustrate our proposed *Attribute Tree* and the effectiveness of *AtTGen* model, as shown in Figure 4. In the first case, the sentence contains an out-of-schema attribute, “sea trialed”, which is ignored by the BERT-based extraction model. While our *AtTGen* model, starting from a given subject, identifies all attribute pairs including the purely literal one by first listing all possible attribute values and then smoothly corresponding to names based on the value and the context. In the other case, the number “158,700” is misextracted as “700” by the Bert-based extractor due to the interference of the thousands-separator. This roots in the model’s failure to really understand numerical values, which is a unique challenge to deep learning-based techniques (Xue et al., 2022). Nonetheless, *AtTGen* directly captures the boundary pattern of numbers and successfully retains the complete value with the span copier, showing a possible solution for this challenge.

6 Related Works

Attribute Extraction is a classical IE task with extensive research. In earlier years, heuristic rules and dictionaries were usually used to iden-

tify attributes and extract attribute values from the texts (Tan et al., 1999; Sasaki and Matsuo, 2000; Vandic et al., 2012; More, 2016; Zheng et al., 2018; Yan et al., 2021). With the development of deep learning for NLP, researchers attempt to leverage neural network technology-based model for tagging attributes (Huang et al., 2015; Hakkani-Tür et al., 2016; Mai et al., 2018) or classifying attribute types (Riedel et al., 2010; Zeng et al., 2014; Amplayo, 2019; Iter et al., 2020; Zhao et al., 2021). Beyond CWA AE, researchers also explore AE in OWA scenario, e.g., some prior works try to expand free attributes from plain texts (Wong and Lam, 2010; Zhang et al., 2022; Cui et al., 2018) and extract the values of schema-free attributes (Xu et al., 2019). Recently, more novel frameworks are proposed to generalize the capacity of AE models. AVEQA (Wang et al., 2020; Shinzato et al., 2022) and MAVEQA (Yang et al., 2022) introduce Question Answering framework for AE task, and Roy et al. (2022) tries to employ large-scale PLM to introduce external knowledge. Further, some academics propose multimodal AE tasks and datasets to enrich the research (IV et al., 2017; Zhu et al., 2020). **Generative Information Extraction**, a rising technique in these two years (Ye et al., 2022), is also an inspiration for proposing this research. A contemporaneous work (Roy et al., 2022) adopts sequence generation models and preliminarily shows the potential of generative models in open-world attribute extraction. Alongside sequence-based generation models, structure generation models are also widely studied and have shown power in other IE tasks. For example, REBEL (Huguet Cabot and Navigli, 2021) introduces a structure-linearized model for relation extraction; Seq2UMTree (Zhang et al., 2020) conducts a sequence-to-unordered-multi-tree generation model for extracting entities and relations jointly; UIE (Lu et al., 2022) proposes a text-to-structure generation framework that can universally model different IE tasks based on the guidance of the pre-defined schema.

Though both attribute extraction and generative models have been widely explored, we are the first to design a novel tree generation model for AE and demonstrate the effectiveness on our unified real-world paradigm.

7 Conclusion and Future Work

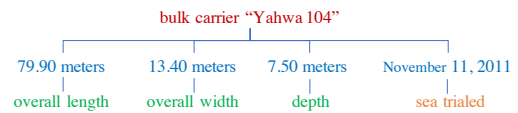
In this paper, we formulate the real-world AE task into a unified *Attribute Tree*, and propose a simple

On November 11, 2011, Nantong Yahwa Shipbuilding’s newly built bulk carrier “Yahwa 104”, with an overall length of 79.90 meters, an overall width of 13.40 meters and a depth of 7.50 meters, was successfully sea trialed. (Chinese version: 2011年11月11日, 南通亚华船舶新造散货船顺利试航, 该散货船“亚华 104”总长 79.90 米、总宽 13.40米、型深 7.50米)

Gold Standard: <overall length, 79.90 meters> <overall width, 13.40 meters>
<depth, 7.50 meters> <sea trialed, November 11, 2011>

BERT: <overall length, 79.90 meters> <overall width, 13.40 meters> <depth, 7.50 meters>

Ours: <overall length, 79.90 meters> <overall width, 13.40 meters> <depth, 7.50 meters>
<sea trialed, November 11, 2011>



The tanker “AST Sunshine” is 274 meters long, 48 meters wide and has a displacement of 158,700 dwt. (Chinese version: 油船“AST Sunshine”全长 274 米, 宽 48 米, 排水量为 158,700 载重吨)

Gold Standard: <overall length, 274 meters> <overall width, 48 meters> <deadweight, 158,700 dwt>

BERT: <overall length, 274 meters> <overall width, 48 meters> <deadweight, 700 dwt>

Ours: <overall length, 274 meters> <overall width, 48 meters> <deadweight, 158,700 dwt>

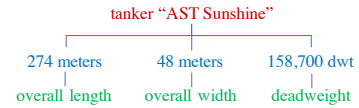


Figure 4: Two cases of the method. The spans in red, blue, green, yellow represent subject entities, attribute values, pre-defined attribute names, and literal attribute names respectively.

but effective tree-generation model to extract both in-schema and schema-free attributes from texts. Experiments on three public datasets demonstrate our prominent performance over various scenarios, and detailed analyses also reveal several interesting findings for attribute extraction.

Several potential directions are left for the future. The first one is that our current approach does not utilize the commonly-provided multimodal information in e-Commerce, which can be naturally introduced into our tree structure as nodes for better results later. Besides, PLM has powerful effects on understanding the semantics of texts and scaling to open-domain AE applications, so incorporating knowledge of different granularity from PLMs is also an attractive extension to be explored.

8 Limitations

Adapting PLMs to our proposed model does not go as smoothly as expected, because there are three different forms of tokenization: the PLM tokenizer, the multilingual tokenizer implemented in our proposed model, and the special annotations of numerical values/entity mentions/long-winded attribute values in the attribute extraction datasets, which are difficult to reconcile simultaneously. Although our model without PLM has outperformed PLM-based ones, this does impose a limitation for future explorations.

Although Re-CNShipNet, one of the datasets used in our experiments, is more accurate with our careful re-annotating, the size of which is still so small that would produce randomness bias during the model training and may affect the final experimental results.

Besides, due to the limitation of computational resources, we did not conduct experiments on large

language models such as T5 (Raffel et al., 2020), LLaMA (Touvron et al., 2023), etc., which may lead to insufficiency of the experiment.

Ethics Statement

This work uses three publicly available datasets, and we respect and adhere to their user agreements and licenses. The content of pre-existing datasets does not reflect our perspectives. We, the in-house authors, re-annotate one of these datasets, i.e., Re-CHShipNet; the purpose of re-annotation is mainly to correct errors and re-balance the ratio of CWA/OWA labels. The annotation may introduce personal judgment and bias, which may bring potential risks. Further, the potential downstream applications of this work include knowledge graph construction, search engine, e-Commerce, recommendation system, etc.; we caution that our proposed method may cause misextraction or false information, and may fail in the case of out-of-distribution and domain shift, which may harm those applications.

Acknowledgements

This work was supported by NSFC under grant 61932001 and U20A20174. Lei Zou is the corresponding author of this paper. We would gratefully appreciate the reviewers for their precious comments that help us to improve this manuscript.

References

Reinald Kim Amplayo. 2019. Rethinking attribute representation and injection for sentiment classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5602–

- 5613, Hong Kong, China. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. [BERT for joint intent classification and slot filling](#). *CoRR*, abs/1902.10909.
- Jason P.C. Chiu and Eric Nichols. 2016. [Named entity recognition with bidirectional LSTM-CNNs](#). *Transactions of the Association for Computational Linguistics*, 4:357–370.
- Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Lei Cui, Furu Wei, and Ming Zhou. 2018. [Neural open information extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413, Melbourne, Australia. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2016. [Language to logical form with neural attention](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.
- Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew E. Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explor.*, 8:41–48.
- Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. [Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM](#). In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 715–719. ISCA.
- 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. *ArXiv*, abs/1508.01991.
- Pere-Lluís Hugué Cabot and Roberto Navigli. 2021. [REBEL: Relation extraction by end-to-end language generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Dan Iter, Xiao Yu, and Fangtao Li. 2020. [Entity attribute relation extraction with attribute-aware embeddings](#). In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 50–55, Online. Association for Computational Linguistics.
- Robert L Logan IV, Samuel Humeau, and Sameer Singh. 2017. Multimodal attribute extraction. In *6th Workshop on Automated Knowledge Base Construction*, Long Beach, California, USA.
- Mayank Jain, Sourangshu Bhattacharya, Harshit Jain, Karimulla Shaik, and Muthusamy Chelliah. 2021. [Learning cross-task attribute - attribute similarity for multi-task attribute-value extraction](#). In *Proceedings of the 4th Workshop on e-Commerce and NLP*, pages 79–87, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yanzeng Li, Bowen Yu, Li Quangan, and Tingwen Liu. 2021. [FITAnnotator: A flexible and intelligent text annotation system](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 35–41, Online. Association for Computational Linguistics.
- Yanzeng Li and Lei Zou. 2022. [gbuilder: A scalable knowledge graph construction system for unstructured corpus](#).
- Bing Liu and Ian R. Lane. 2016. [Attention-based recurrent neural network models for joint intent detection and slot filling](#). In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 685–689. ISCA.
- Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. [Tree-structured decoding for solving math word problems](#). In *Proceedings of the*

- 2019 *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2370–2379, Hong Kong, China. Association for Computational Linguistics.
- Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. [Unified structure generation for universal information extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.
- Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. [Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.
- Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. [An empirical study on fine-grained named entity recognition](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Per Martin-Löf. 1996. On the meanings of the logical constants and the justifications of the logical laws. *Nordic journal of philosophical logic*, 1(1):11–60.
- Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *ArXiv*, abs/1608.04670.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Sujith Ravi and Marius Pasca. 2008. Using structured text for large-scale attribute extraction. In *International Conference on Information and Knowledge Management*.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.
- Kalyani Roy, Tapas Nayak, and Pawan Goyal. 2022. [Exploring generative models for joint attribute value extraction from product titles](#). *CoRR*, abs/2208.07130.
- Yutaka Sasaki and Yoshihiro Matsuo. 2000. [Learning semantic-level information extraction rules by type-oriented ILP](#). In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.
- Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2022. [Simple and effective knowledge-driven query expansion for QA-based product attribute extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 227–234, Dublin, Ireland. Association for Computational Linguistics.
- Ah-Hwee Tan et al. 1999. Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases*, volume 8, pages 65–70.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Damir Vandić, Jan-Willem Van Dam, and Flavius Frasincar. 2012. Faceted product search powered by the semantic web. *Decision Support Systems*, 53(3):425–437.
- Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. [Learning to extract attribute value from product via question answering: A multi-task approach](#). In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.
- Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. [A novel cascade binary tagging framework for relational triple extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, Online. Association for Computational Linguistics.
- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.
- Tak-Lam Wong and Wai Lam. 2010. [Learning to adapt web information extraction knowledge and discovering new attributes via a bayesian approach](#). *IEEE Transactions on Knowledge and Data Engineering*, 22(4):523–536.
- Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. [Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.
- Bingcong Xue, Yanzeng Li, and Lei Zou. 2022. Introducing semantic information for numerical attribute prediction over knowledge graphs. In *The Semantic Web – ISWC 2022*, pages 3–21, Cham. Springer International Publishing.

- Jun Yan, Nasser Zalmout, Yan Liang, Christan Grant, Xiang Ren, and Xin Luna Dong. 2021. [AdaTag: Multi-attribute value extraction from product profiles with adaptive decoding](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4694–4705, Online. Association for Computational Linguistics.
- Li Yang, Qifan Wang, Zac Yu, Anand Kulkarni, Sumit Sanghai, Bin Shu, Jon Elsas, and Bhargav Kanagal. 2022. [Mave: A product dataset for multi-source attribute value extraction](#). In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22*, page 1256–1265.
- Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. [Generative knowledge graph construction: A review](#). *CoRR*, abs/2210.12714.
- Bowen Yu, Zhenyu Zhang, Jiawei Sheng, Tingwen Liu, Yubin Wang, Yucheng Wang, and Bin Wang. 2021. Semi-open information extraction. In *Proceedings of the Web Conference 2021*, pages 1661–1672.
- Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proc. of ECAI*.
- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. [Relation classification via convolutional deep neural network](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Li Zhang, Yanzeng Li, Rouyu Zhang, and Wenjie Li. 2021. [Semi-open attribute extraction from chinese functional description text](#). In *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 1505–1520. PMLR.
- Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. [Minimize exposure bias of Seq2Seq models in joint entity and relation extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.
- Xinyang Zhang, Chenwei Zhang, Xian Li, Xin Luna Dong, Jingbo Shang, Christos Faloutsos, and Jiawei Han. 2022. [Oa-mine: Open-world attribute mining for e-commerce products with weak supervision](#). In *Proceedings of the ACM Web Conference 2022*, pages 3153–3161.
- Jiapeng Zhao, Panpan Zhang, Tingwen Liu, Zhenyu Zhang, Yanzeng Li, and Jinqiao Shi. 2021. [Relation extraction based on data partition and representation integration](#). In *2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*, pages 68–75.
- Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. [Opentag: Open attribute value extraction from product profiles](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1049–1058. ACM.
- Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. [Attention-based bidirectional long short-term memory networks for relation classification](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.
- Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2018. [Sequential copying networks](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4987–4995. AAAI Press.
- Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. [Multimodal joint attribute prediction and value extraction for E-commerce product](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2129–2139, Online. Association for Computational Linguistics.

A Implementation Details

We implement our model on PyTorch, and manually tune the hyper-parameters based on the dev set. It is trained using Adam with the batch size/learning rate/maximum training epoch set to 512/0.0002/40. The model of the best epoch evaluated on the dev set is saved as the final model. For the encoder, we use 200-dimensional embeddings; the 2-layer BiLSTM_{enc} is configured with 200 hidden state size, and the kernel size of Conv_{enc} is set to 3. For the decoder, we use a 1-layer unidirectional LSTM_{dec} for decoding the state, and Conv_{dec} with the same configuration of Conv_{enc} to extract the generative features. All the experiments are performed on a cluster with Nvidia A40 GPUs, and we run each experiment 5 times with different seeds, reporting the average scores to ensure reliability. For more implementation details, please refer to our publicly available repository at <https://github.com/lsvih/AtTGen>.