Subsumption Prediction for E-Commerce Taxonomies

Jingchuan Shi[1(✉)], Jiaoyan Chen[1], Hang Dong[1], Ishita Khan[2], Lizzie Liang[2], Qunzhi Zhou[2], Zhe Wu[2], and Ian Horrocks[1]

[1] Department of Computer Science, University of Oxford, Oxford, UK
{jingchuan.shi,jiaoyan.chen,hang.dong,ian.horrocks}@cs.ox.ac.uk
[2] eBay Inc., San Jose, USA
{ishikhan,lizliang,qunzhou,zwu1}@ebay.com

Abstract. Taxonomy plays a key role in e-commerce, categorising items and facilitating both search and inventory management. Concept subsumption prediction is critical for taxonomy curation, and has been the subject of several studies, but they do not fully utilise the categorical information available in e-commerce settings. In this paper, we study the characteristics of e-commerce taxonomies, and propose a new subsumption prediction method based on the pre-trained language model BERT that is well adapted to the e-commerce setting. The proposed model utilises textual and structural semantics in a taxonomy, as well as the rich and noisy instance (item) information. We show through extensive evaluation on two large-scale e-commerce taxonomies from eBay and AliOpenKG, that our method offers substantial improvement over strong baselines.

Keywords: Subsumption Prediction · E-Commerce Taxonomy · Pre-trained Language Model · BERT

1 Introduction

Taxonomies capture the *is-a* relationships between concepts, facilitating their storage, classification and organisation [4]. In e-commerce, taxonomy provides the basis for item categorisation, and is vital for search, inventory management and recommendation. Most e-commerce sites support two methods for users to locate a product: category browsing and keyword search. For the former, the taxonomy itself is presented to the user to navigate; for the latter, the taxonomy also provides important information to the search engine, which usually attempts to narrow the range of search results down to one or a few categories before retrieving and ranking items. For item recommendation, placement in the taxonomy is one of the most important heuristics in relevance scoring [31]. As such, the completeness and accuracy of the taxonomy has a major impact on sales and user experience.

Taxonomy-related research mainly includes taxonomy construction, curation and applications. These tasks have a close bond with natural language processing (NLP) and ontology engineering [13], the latter of which studies similar

C. Pesquita et al. (Eds.): ESWC 2023, LNCS 13870, pp. 244–261, 2023.
https://doi.org/10.1007/978-3-031-33455-9_15

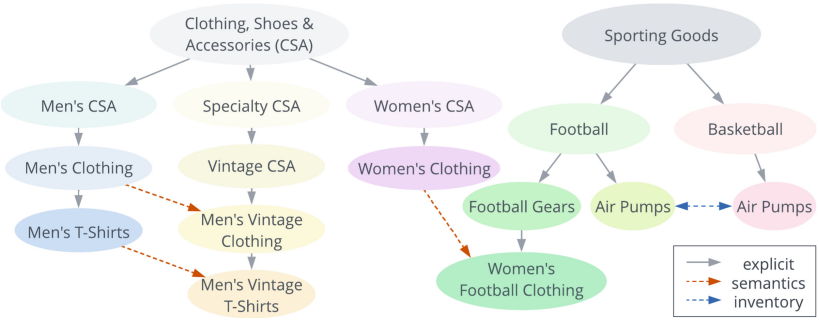

Fig. 1. Example of missing subsumptions in the e-commerce taxonomy

abstractions but typically involves more complex representations and utilises logical reasoning. Many taxonomies start as lightweight catalogues that can simply be curated by hand. However, as numerous taxonomies are constantly being created and existing ones constantly expanded, these tasks often become very labour intensive, and therefore their automation has become an important research topic. Among these tasks, subsumption prediction concerns adding new *is-a* relations between concepts, and is a major component of taxonomy curation.

The task of subsumption prediction is challenging. When taxa have complex, multifaceted semantics (e.g., e-commerce categories), the taxonomies are usually constructed in a way that each level specifies one or several facets (e.g., brand, material, function, etc.) on top of the parent category [18]. Theoretically, the order at which some facets are specified can be interchangeable, with no influence on the class's overall semantics. This leads to one kind of missing subsumption. For instance in Fig. 1, the categories *Men's Vintage Clothing* and *Men's Vintage T-Shirts*[1] should be considered subcategories of, respectively, *Men's Clothing* and *Men's T-Shirts*,[2] because all vintage clothing is clothing and all vintage t-shirts are t-shirts. Similarly, *Women's Football Clothing* is in reality a subcategory of *Women's Clothing*. However, these subsumptions may not be recognised because the categories belong to different branches in the hierarchy, although the two branches actually converge to a significant extent. A corollary of this observation is that while many taxonomies are organised into trees, the branches of these trees are not necessarily mutually exclusive; in our example there is some overlap between "Specialty CSA" and "Men's CSA".

While many missing subsumptions can be found by analysing the semantics of class labels, the underlying item level information could also be helpful. Each category in an e-commerce taxonomy is not only an abstract taxon, but also a label for a collection of inventory items. In Fig. 1, it is easy to judge from the class labels that *Football Air Pumps* and *Basketball Air Pumps*[3] are similar categories,

[1] Browse this category and the taxonomy around it at https://www.ebay.com/b/175781.
[2] https://www.ebay.com/b/15687.
[3] https://www.ebay.com/b/261761 and https://www.ebay.com/b/261791.

but it is not obvious that such pumps are compatible with each other and that the two categories should thus be mutually subsuming (i.e., equivalent). Discovering this kind of subsumption might be possible using a statistical approach based on the very large (and noisy) sets of relevant inventory items. However, it is unclear how to integrate the semantic understanding of category labels with the information from items.

Related Works. There is a large body of work on related areas, most prominently knowledge graph (KG) link prediction and taxonomy enrichment. KG link prediction is concerned with predicting relational facts (e.g., (*France*, *hasCapital*, *Paris*)) [29,30], often utilising different kinds of KG embedding models such as TransE [3], DistMult [38], and HolE [26]. However, these methods aim at relational facts, which can be understood as a multi-relation graph. They are not directly applicable in our e-commerce taxonomy curation given the taxonomy's noisy, multi-faceted, and hierarchical nature. Taxonomy enrichment [15] mines new concepts from a corpus and adds them to a taxonomy. Some enrichment methods can perform subsumption prediction for e-commerce taxonomies. Octet [24] is a two-stage pipeline that tackles edge prediction by applying a feed forward NN over features obtained from graph embeddings, word embeddings, and lexical metrics such as edit distance. While the model has achieved major improvements over non e-commerce specialised baselines, it uses non-contextual word embeddings, which leaves much room for improvement. AliCoCo [23] builds a massive, multi-layered KG of e-commerce concepts and links the concepts with items.

Another closely related field is ontology curation using deep learning. While numerous ontology embeddings such as OPA2Vec [34] and OWL2Vec* [6] can be applied to predict subsumptions, the amount of work focusing on optimising subsumption prediction is limited. BERTSubs [5] utilises BERT [9], a pre-trained language model (PLM) that has been shown to produce high quality contextual embeddings, and applies templates to convert candidate subsumptions into sentences for classification with BERT. The BERT is then attached to a classifier layer, and jointly fine-tuned using existing subsumptions. Evaluation on ontologies shows that BERTSubs can dramatically outperform early KG link prediction methods such as TransE and DistMult. However, BERTSubs ignores the aforementioned characteristics of e-commerce taxonomies, especially the existence of items.

In this work, we propose a new subsumption prediction approach that enhances previous work, taking into consideration the noisiness and richness of e-commerce taxonomies. Our approach features 1) BERT-based contextual embeddings with carefully designed templates; 2) a pipeline based on existing NLP tools to leverage lexical semantics in class labels; and 3) utilisation of instance data (i.e., product items). The practically optimal usage of BERT has been a long standing problem for researchers [22]. Our solution to this problem with templates and preprocessing proves to work well for subsumption prediction, and can generalise to other tasks of a similar nature. We propose two ways to combine instances with class label semantics, i.e., attention-based and

template-based. Most BERT-based classification models use a feed forward neural network classifier that applies to a fixed amount of embedding vectors to make one prediction. In our instance-aware model, every prediction has to be based on a variably-sized set of embedding vectors, with one vector corresponding to one instance. Therefore, we also study two alternative classifiers besides feed forward layer: box embedding and extensional inference via k-nearest neighbours.

We evaluate our method on two large scale e-commerce taxonomies from eBay and AliOpenKG[4]. Both taxonomies are equipped with millions of items. Experiments have verified the effectiveness of the sentence processing pipeline and the consideration of items (instances), and clearly demonstrate that our solutions on path text preprocessing, templates and fine-tuning can help realise the full potential of BERT. We summarise our main contributions as follows:

1. Propose a taxonomy subsumption prediction framework based on contextual representations and also able to exploit instance data.
2. Extensively evaluate our framework and associated techniques on two e-commerce taxonomies.

2 Preliminaries

2.1 Pre-trained Language Model BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers [9], is a transformer-based pre-trained language model (PLM) for contextual representations. It consists of a stack of encoder units (12 units in the original release `bert-base`) and self-attention heads. It is usually pre-trained with large, general-purpose corpora to learn sufficient understanding of the language itself. BERT is used in conjunction with a tokeniser based on WordPiece [32], where a single word may be split into multiple sub-word tokens, e.g. *stainless* into "`stain`" and "`##less`". The original BERT model is pre-trained on two tasks: masked language modelling (MLM) and next sentence prediction (NSP) [9]. MLM aims to predict some randomly masked tokens in the sentences, while NSP is to predict the following sentence of a given sentence. For a given sentence, standalone BERT can produce the contextual embedding of each individual token, as well as the embedding of the entire sentence, which is the embedding of a special token `[CLS]` added in front of the sentence.

An effective and popular way of applying BERT for downstream tasks is attaching an additional neural layer, and fine-tuning both BERT and the additional layer w.r.t. a task-specific loss and given samples. For classification, the textual input is either "`[CLS]` Sentence" for tasks on a single sentence, or "`[CLS]` Sentence A `[SEP]` Sentence B" for tasks on sentence pairs (`[SEP]` is a special token for separating two sentences). In our work, we adopt this fine-tuning paradigm as it allows BERT to adapt to the task's peculiarities, i.e. uncommon input and/or specialised classification objective.

[4] Code and data available at https://github.com/jingcshi/bert_subsumption.

2.2 Box Embeddings

The transitive, asymmetric nature of subsumption prohibits usage of symmetric similarity measures, e.g. Cosine similarity and Euclidean distance, when making predictions based on embeddings such as those produced by BERT. On the other hand, some geometric embeddings [10,27,28,36] have a natural ability to express subsumption and are thus suitable for embedding taxonomies. Box embeddings [37] have recently received attention as an effective taxonomic embedding method, where classes are mapped to high dimensional boxes and subsumption naturally translates to box containment. Informally, one may think of box embeddings as high dimensional Venn diagrams.

The original box embedding is a lattice structure in $\mathbb{R}^d$. A box is defined as $x = (x^m, x^M)$ where $x^m, x^M \in \mathbb{R}^d$ are the lower bound and upper bound coordinates, respectively. For two boxes x and y, the intersection $x \wedge y$ is naturally defined as their geometric intersection. The volume of a box x is $|x| = \prod_i (x_i^M - x_i^m)$. In order to learn embedding parameters from known subsumptions, a naïve loss function is to maximise proportional overlap:

$$\mathcal{L}(x, y) = \log |x \wedge y| - \log |x| \tag{1}$$

In other words, the ideal box embedding should have all child boxes completely submerged in parent boxes. Unfortunately, this loss function leads to poor performing models due to unbounded gradient and unfavorable local minima that hinder optimisation [8,19]. Therefore, numerous approaches have been investigated to soften the box boundaries [20] by redefining the box as a probabilistic distribution along each dimension. In this paper, we adopt the state-of-the-art soft box embedding to our knowledge, dubbed GumbelBox [8], that defines boxes as multi-dimensional Gumbel variables [12]. Key merits of Gumbel distribution are that 1) the max of two Gumbel variables with the same scale parameter β is another Gumbel variable, therefore with careful definition one can assure that the intersection of two Gumbel boxes is another Gumbel box; 2) it is smooth and mildly skewed, resulting in easier gradient descent.

3 Problem Statement

We define a *taxonomy* to be a set $T = (\mathcal{C}, \mathcal{R})$, where $\mathcal{C}$ is a set of classes and $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is a set of *is-a* relations. By definition, $\mathcal{R}$ is a partial ordering over $\mathcal{C}$, thus is transitive. We say C subsumes D if $(D, C) \in \mathcal{R}$ or (D, C) can be entailed from $\mathcal{R}$ via transitivity, and $C \equiv D$ if C subsumes D and D subsumes C. Note that many class hierarchies defined in OWL ontologies [2] can be regarded as a kind of taxonomy. The subsumption (is-a) relation between two OWL classes is defined by a built-in property *rdfs:subClassOf*.

We assume every class in the taxonomy has a text label and a set of instances which can be empty: $C = (l, P)$. In the e-commerce taxonomy we investigate, instances are items which are represented by a variety of modalities: text label, image, property-value pairs, etc. For the time being, we only consider the text

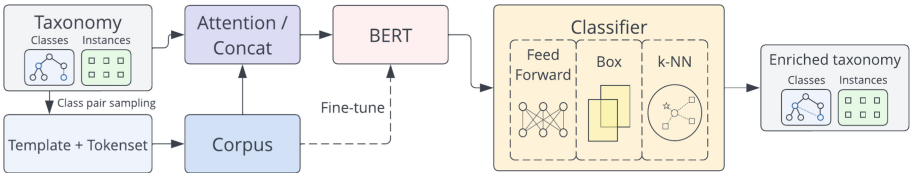

Fig. 2. Framework overview: (i) corpus construction, (ii) fine-tuning, (iii) joint embedding of label paths and instances, (iv) subsumption prediction using feed forward layer/box embedding/extensional inference with k-NN

modality and thus every $p_i \in P$ as well as every l_i are strings. We define the *label path* of a class to be the sequence of labels for all its ancestor classes and itself: If $C_0 \supseteq C_1 \supseteq \ldots \supseteq C_i$ is the longest possible subsumption chain for C_i, then the label path ("path" for simplicity) $\overline{l_i} = (l_0, l_1, \ldots, l_i)$. While the individual label for a class often misses important information, the path provides a more complete yet redundant textual description. The label for *Men's Vintage T-Shirts* is in fact just "T-Shirts" which is indistinguishable from many other classes, while the full path "Clothing, Shoes & Accessories → Specialty → Vintage → Men's Vintage Clothing → T-Shirts" has duplicate occurrences of "Clothing" and "Vintage", and one may consider the co-occurrence of "Clothing" and "T-Shirts" to be another type of redundancy since the former is a hypernym of the latter.

We model taxonomy subsumption prediction as a binary classification problem: given two classes $C_1 = (l_1, P_1)$ and $C_2 = (l_2, P_2)$ in a taxonomy $(\mathcal{C}, \mathcal{R})$, a score $s \in [0, 1]$ indicating the likelihood of $C_1 \subseteq C_2$ is expected. Subsumption can be interpreted in two ways, either intensionally or extensionally [7,35]: (i) intentionally, the abstract class defined by the semantics of $\overline{l_2}$ encompasses that of $\overline{l_1}$; (ii) extensionally, P_2 encompasses P_1 in the sense that any instance of P_1 either appears directly in P_2, or has a sufficiently close neighbour in P_2.

4 Methodology

4.1 Framework Overview

Our framework, as shown in Fig. 2, operates within the standard paradigm of BERT fine-tuning for classification. Given a taxonomy, we start by constructing the training corpus, consisting of instances and preprocessed label paths. A fast and simple label preprocessing technique (the process Tokenset) is used both as guidance for negative sampling and as a step in preparing the corpus. Next, we fine-tune the BERT model using existing positive subsumptions and negative subsumptions extracted from the previous step. We then use the trained BERT to obtain either instance embeddings when instances are present, or path embeddings, when instances are not present or not to be used. We ensure that instance embeddings contain class label semantics via either an attention mechanism or a concatenation template. Lastly, we convert the embeddings to subsumption

predictions with one of three classifiers: feed forward layer, box embedding, and extensional inference with k-nearest neighbours.

4.2 Sample Construction

The goal of this stage is to obtain two sets of class pairs, one positive and one negative. The model first stores all the known subsumptions in a sparse matrix, and constructs the label path for each class. Since any two classes could potentially be a missing subsumption, it is difficult to safely sample negative pairs without hitting false negatives. To circumvent this problem, we designed Tokenset, a process that flattens out labels or paths into a list of keywords, allowing us to apply a filtering heuristic. Tokenset uses WordNet [25] to tokenise sentences, give part-of-speech tags to tokens and lemmatise them, as well as to identify and delete the hypernym in any hypernym-hyponym pair present in the list, thus removing the main source of noise in paths. An example of tokenset construction is the *Men's T-Shirt* class, whose label path is "Clothing, Shoes & Accessories → Men → Men's Clothing → Shirts → T-Shirts". Tokenset first produces the set {clothing, shoe, accessory, men, shirt, t-shirt}, then removes clothing and shirt in the hypernym reduction phase since both terms are hypernyms of t-shirt. The final representation {men, accessory, t-shirt, shoe} contains all the relevant information, i.e., men and t-shirt, and is much more concise. *Accessory* and *shoe* were not removed by this process because they are the irrelevant terms in a three way disjunction ("Clothing, Shoes & Accessories" really means "clothing" ∨ "shoes" ∨ "accessories"), and deletion of such terms would require construction of logical expressions from natural language, which adds further complexities; we leave this for future study.[5]

Once we obtain the abridged tokensets for two classes, we count the number of unique tokens that appear in one set but not the other. Such tokens usually represent semantic constraints that are unique to one of the classes. Taking into account the case where a tokenset may contain a hyponym of the other set's token, we apply the following criterion for negative sampling:

$$|\mathsf{Tokenset}(\overline{l_1} \cup \overline{l_2})| - \max\left(|\mathsf{Tokenset}(\overline{l_1})|, |\mathsf{Tokenset}(\overline{l_2})|\right) > 2 \tag{2}$$

The equation demands that both tokensets contain at least three elements that are unique to themselves. It is very unlikely that classes satisfying this condition have a subsumption relationship.

We use all direct positive subsumptions as the positive set. For every positive pair (C_1, C_2), we replace C_2 with a random class and add the resulting pair to the negative set if the above negative sampling criterion is satisfied.

[5] The difficulty of such task is illustrated by labels featuring a mixture of conjunction and disjunction, e.g., *Suit Jackets & Blazers*, which means "Suit Jackets" ∨ "Blazers", and *North & Central America*, which means "North America" ∨ "Central America".

4.3 Corpus and Fine-tuning

The tokenset representation obtained above is close to the actual training corpus. In order to facilitate BERT's understanding of this unusual type of input, we place a fixed template "**A category of products defined by:**"[6] before the tokenset. The final input for *Men's T-Shirts* is therefore "**A category of products defined by: t-shirt, men, accessory, shoe**". The phrasing of this template is empirical, but the idea is to formulate the keyword-like taxonomic path as natural language. Evaluation results in the next section make it clear that such templates improve performance dramatically. A speculative explanation is that these templates establish the context for BERT so that it reads the following tokenset as related to categorisation, rather than as a random collection of words. A recent prompt learning study [17] reveals a similar phenomenon on GPT-3, another large PLM. It may be possible to improve the proposed templates by fine-tuning the embedding of the template tokens, similarly to learning a soft prompt [21]; we leave this for future study.

We denote the templated tokenset for a class C_i as $\overline{l_i}^*$. The actual fine-tuning task adds a feed forward layer on top of BERT as a classifier head for binary classification. For a pair (C_1, C_2), we feed the standard classification template `[CLS]` $\overline{l_1}^*$ `[SEP]` $\overline{l_2}^*$ into the model. In fine-tuning, we shuffle the order of tokens within each tokenset to prevent the model from simply learning to match exact sequences. The classifier uses the output of `[CLS]` token, a 768-dimensional vector as input, and returns a score $s \in [0, 1]$ indicating the predicted subsumption likelihood. We use cross entropy loss over the corpus as the training objective, and fine-tune the BERT and the classifier jointly using an Adam optimiser [16]. We do not use instance data for fine-tuning because most instance data are noisy and have poorer quality compared to the human curated class labels.

4.4 Prediction

Prediction Without Instances. When instances are not present or are disabled, the model makes predictions with the feed forward NN classifier fine-tuned in Sect. 4.3. A broad list of candidate class pairs is either externally provided or manually generated. There is no single best method of generating candidate pairs, and we generate by ranking nearest neighbours w.r.t. Euclidean distances between the classes' embeddings. Note that the class embedding is the `[CLS]` token output of the fine-tuned BERT given the class's preprocessed tokensets. For each candidate, we take the output of the classifier directly as prediction.

Prediction with Instances. The prediction process is very different when instances are involved. First, the model input is no longer templated tokensets but instance labels. The model yields an embedding for each instance, where we

[6] For the Chinese AliOpenKG dataset (see Sect. 5.1), the template is "产品类目: ".

apply either of the following two techniques to ensure the presence of class label information in item embeddings:

- **Attention.** For an instance $p = t_1 t_2 \dots t_k$ in a class with path $\bar{l}$, we first separately compute the sentence embedding of $\bar{l}^*$, denoted $\boldsymbol{e}_l$, and the contextual individual token embeddings of $t_1, t_2, \dots, t_k$, denoted $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_k]^T$. The final instance embedding is a weighted sum of individual embeddings, where the weights are given by a softmax over the dot products between $\boldsymbol{e}_i$ and $\boldsymbol{e}_l$:

$$\boldsymbol{e} = \text{softmax}(\frac{\boldsymbol{e}_l \boldsymbol{E}^T}{\sqrt{d}})\boldsymbol{E} \tag{3}$$

  where $d = 768$ is the embedding dimension. As an example, Fig. 3 shows the learned attention weights for the item titled "Original Kawasaki T-Shirt Iron-On Vintage 70s UNUSED Transfer" w.r.t. the category *Men's Vintage T-Shirts*. Attention places a higher weight on tokens closer to the label-based embedding, and consequently the instance embeddings gravitate towards it.
- **Concatenation.** Another approach is to concatenate the instance and label using the template "`[CLS]` Item: p in the category defined by: $\bar{l}^*$", which may vary for different taxonomies. For instance, we use "`[CLS]`产品名: p, 类目: $\bar{l}^*$" for the AliOpenKG dataset.

Token	Original	Kawasaki	T	-	Shirt	Iron
Weight	.0975	.0766	.0733	.0684	.0983	.0781
Token	-	On	Vintage	70s	UNUSED	Transfer
Weight	.0704	.0836	.1060	.0830	.0843	.0805

Fig. 3. Attention weights of tokens in an item title given the preprocessed label path "*vintage, accessory, t-shirt, men, specialty, shoe*".

After we embed each instance for both classes, we effectively have two vector clusters and the goal is to decide if one cluster encompasses the other. One of the following three classifier heads is used to obtain a prediction:

- **Feed forward.** As in the case without instances, we use the BERT with feed forward classifier. However, since the object is no longer a single sentence, we perform a simple but widely used ensemble technique, i.e., for each instance in the subclass, we pair it with a random instance in the superclass, predict each pair and report the average score.
- **Box embedding.** The motivation for using box embedding is to "draw a box around the cluster" and leverage the geometric properties of boxes. However, the 768 dimensional space where BERT outputs reside is not suitable for box embeddings, as volume and intersection become extremely unstable at such a

high dimension. Moreover, the vector clusters are often anisotropic and have irregular shapes. As such, we train a multilayer perceptron (MLP) that uses the means and standard deviations of a cluster along each dimension as input features, and projects this 1,536 dimensional feature to a low dimensional GumbelBox embedding [8]. The training data for this MLP is based on the same corpus for fine-tuning, with the label paths replaced by pre-computed features for each class. We employ the KL-divergence loss and use an Adam optimiser. The proportional box overlap $\frac{|\boldsymbol{c}_1 \wedge \boldsymbol{c}_2|}{|\boldsymbol{c}_1|}$ is reported as the final score of the subsumption (C_1, C_2), where $|\cdot|$ measure the volumn of a box, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ denote the boxes of C_1 and C_2, respectively.

– **Extensional inference with k-NN.** In the problem statement, we interpreted a possibility for subsumption as the instances of the parent encompassing those of the child. We call this approach extensional inference as the idea can trace its roots to the logical definition of hyponymy, where a hypernym tends to have wider extension (or more objects) than its hyponym [7,11,35]. We can thus find a concrete formulation of such idea in the context of two vector clusters of C_1 and C_2, given as:

$$C_1 \subseteq C_2 \;\leftrightarrow\; \forall x \in V(C_1). \exists\, y \in V(C_2).\; (\|x - y\| \leq d_0) \tag{4}$$

where $V(\cdot)$ denotes the vectors of a class. In other words, all instances in the subclass's cluster are within distance d_0 from some instance in the superclass's cluster in embedding space, for some constant d_0 to be determined empirically. When the sizes of clusters are not formidably large, it is possible to chase this definition directly, and the resulting algorithm is a k-nearest neighbours (k-NN) with $k = 1$. Perfect containment is rarely feasible in reality, so we change the universal quantifier in the definition above to measuring the percentage of subclass instances that have sufficiently close neighbours in the superclass, reporting this percentage as the prediction score. Existing similarity searching libraries like Faiss [14] can speed up k-NN computation.

This bruteforce approach can be seen as an upper bound for any prediction method based on vector clusters. While it is slow and consumes massive memory storing all the vectors, it utilises full, uncompromised information from the instances, whereas box embedding loses significant information when downsampling the $n \times 768$ vector cluster to a 2×768 feature. Our expectation in evaluation is therefore not for the box embedding approach to beat k-NN, but to approximate k-NN to a sufficient degree.

5 Evaluation

5.1 Datasets and Setup

We conduct experiments on two taxonomies, the eBay taxonomy and the taxonomy extracted from the AliOpenKG ontology[7]. Since AliOpenKG stores the

[7] https://ali.openkg.org/.

classes and instances separately as TBox and ABox, we take the subset of TBox formed by product categories under *rdfs:subClassOf*, and link it with the subset of ABox formed by the label and category membership for each item. We publish this extracted dataset for benchmarking in other e-commerce curation tasks. Table 1 lists basic statistics for the two datasets. Note that both taxonomies in their original form are trees, therefore the number of direct subsumptions is one less than the number of classes.

Table 1. Metadata of the taxonomies used in evaluation

Taxonomy	#Classes	Max depth	Avg. depth	#Instances	Language
eBay	16,888	6	4.223	6.4M	English
Alibaba	7,100	4	3.896	3.1M	Chinese

Task. Manual labelling of subsumptions is difficult and expensive. Therefore, we evaluate by predicting masked subsumptions in the taxonomy. We hold out 10% of the direct subsumptions for testing, 10% for validation and use the remaining 80% for training. Note that membership of instances is inherited along the hierarchy, meaning that the non-leaf classes will automatically include instances from all their descendants, obtained by transitive closure of the direct subsumptions. We sample instance memberships prior to masking. To reflect masking and avoid data leakage, we remove some memberships and truncate some paths accordingly.

Metrics. We report results for mean reciprocal rank (MRR), hits@5 (H@5), precision (P) and recall (R). For each testing or validating subsumption (C_1, C_2), we create a set of negative subsumptions by replacing C_2 with false subsumers. False subsumers come from two sources:

1. Random classes that pass the Tokenset negativity test, which serve as the easy negatives.
2. Taxonomic neighbours. We consider the taxonomy as a graph and enumerate C_2's distance-1 and distance-2 neighbours. These classes will be the grandparents, parents, siblings, children and grandchildren of C_2. We select random classes from this pool and add them to the negative set if the selected class does not subsume C_1 in the original taxonomy. This process is repeated until either n negatives are found or the pool is exhausted, in which case we consider the distance-3 neighbors, then distance-4 neighbours, etc. We set $n = 20$ for both datasets. These classes serve as the hard negatives.

We maintain a 1:1 ratio of easy and hard negatives. The ranking set for each subsumption is therefore $\{C_2, C_{\text{neg}1}, \ldots, C_{\text{neg}2n}\}$ with a size of 41. To calculate P and R, we set a prediction threshold for each model by optimising F1 on the validation set, and apply the threshold to prediction scores on ranking sets.

Model. The eBay dataset is processed with `bert-base-cased`, while the Alibaba dataset is processed with `bert-base-chinese`. Both models can be found at

HuggingFace Transformers[8], and both are proven to have strong understanding of generic day-to-day language.

Baselines. Our set of baselines include three well established ontology embeddings, Onto2Vec [33], OPA2Vec [34], and OWL2Vec* [6], along with BERTSubs, a recent strong-performing subsumption prediction framework based on BERT fine-tuning [5]. Note that disabling instance data, path template and tokenset preprocessing from our method makes it effectively equivalent to the Path Context (PC) variant of BERTSubs for intra-ontology named subsumption prediction; further disabling paths, i.e., working with single class labels, is equivalent to the Isolated Class (IC) variant. Both methods use the output of the `[CLS]` token to represent a sequence and feed it to a classifier. Template formulations are identical in the case of IC, and differ minimally in the case of PC. We also take well-established ontology embeddings, including Onto2vec, OPA2Vec, and OWL2Vec*, which are all based on ontology tailored non-contextual word embeddings, as baselines.

Implementation. We ran all the experiments on a 6-core Intel Core i9 computer with 1x Tesla V100 GPU. We ran 5 epochs for fine-tuning with a learning rate of 5×10^{-5}, and 30 epochs for box embedding training with a learning rate of 2×10^{-5}. The MLP in box embedding has 4 hidden layers with a total of 1.9×10^5 trainable parameters, producing 24 dimensional box embeddings.

5.2 Evaluation Results

Table 2 presents the results of our models and the baselines on predicting masked subsumptions in the eBay taxonomy. A common characteristic of all rows is that recall is significantly higher than precision, which is a consequence of the overwhelmingly negative ranking set. To optimise F1, the prediction thresholds are often set quite low. Consistent with [5], BERTSubs has a significant edge over ontological embedding baselines, showing the superiority of contextual word embedding by BERT. Unsurprisingly, the path-only variant outperforms BERTSubs-IC and BERTSubs-PC, thanks to the template introduced in Sect. 4.3 and the Tokenset pipeline.

Adding instances results in little if any improvement when the feed forward ensemble is used as the classifier. However, the benefit of instances gets pronounced when the classifier is designed for vector clusters. Both box embedding and k-NN show promising results, and box embedding is able to close the gap with the bruteforce approach to 2%, while running much more quickly and being more generalisable (more detailed account on inference speed in Sect. 5.3). Both box embedding and k-NN substantially outperform the feed forward classifier for two reasons. First, while BERT has been pre-trained and can handle item titles, the feed forward layer has not been trained with titles and should not be expected to perform well on them. Second, the ensemble mechanism does not

[8] https://huggingface.co/bert-base-cased and https://huggingface.co/bert-base-chinese.

Table 2. Results of predicting masked subsumptions of the eBay taxonomy

Method	Feed forward					Box embedding					k-NN				
	P	R	F1	MRR	H@5	P	R	F1	MRR	H@5	P	R	F1	MRR	H@5
Onto2Vec	.135	.709	.227	.265	.357	.166	.748	.272	.321	.414	.176	.754	.285	.335	.457
OPA2Vec	.160	.732	.263	.308	.401	.182	.781	.295	.347	.437	.189	.776	.304	.359	.462
OWL2Vec*	.174	.733	.281	.326	.436	.200	.772	.318	.369	.483	.207	.785	.328	.381	.495
BERTSubs-IC	.382	.869	.531	.557	.714	N/A					N/A				
BERTSubs-PC	.197	.840	.319	.493	.625	N/A					N/A				
Ours: *P*	.544	**.872**	.670	.601	.768	N/A					N/A				
Ours: *P+I (att)*	.463	.835	.596	.552	.729	.502	.854	.632	.611	.765	.585	.791	.673	**.633**	.783
Ours: *P+I (con)*	.456	.862	.596	.555	.736	.493	.840	.621	.618	.758	**.588**	.810	**.681**	.629	**.786**
Legend	*P*: path, *I*: instances, *att*: attention, *con*: concatenation														

Table 3. Results of predicting masked subsumptions of the Alibaba taxonomy

Method	Feed forward					Box embedding					k-NN				
	P	R	F1	MRR	H@5	P	R	F1	MRR	H@5	P	R	F1	MRR	H@5
Onto2Vec	.140	.658	.231	.223	.296	.137	.664	.227	.228	.296	.142	.696	.236	.228	.312
OPA2Vec	.151	.698	.248	.246	.327	.153	.689	.250	.245	.311	.155	.715	.254	.249	.333
OWL2Vec*	.189	.742	.301	.284	.380	.194	.736	.307	.290	.393	.199	.721	.311	.300	.408
BERTSubs-IC	.397	.796	.529	.468	.540	N/A					N/A				
BERTSubs-PC	.359	.783	.492	.432	.519	N/A					N/A				
Ours: *P*	.454	.806	.580	.503	.636	N/A					N/A				
Ours: *P+I (att)*	.485	.834	.613	.540	.667	.518	.828	.637	.562	.693	.532	.830	.648	**.583**	**.715**
Ours: *P+I (con)*	.480	**.838**	.610	.532	.656	.520	.831	.640	.569	.704	**.534**	.829	**.650**	.580	.713
Legend	*P*: path, *I*: instances, *att*: attention, *con*: concatenation														

fully capture the essence of subsumption in the vector cluster context, as defined by Eq. (4). The best results are achieved with k-NN on the attention variant, but otherwise attention and concatenation remain close in efficacy.

Results on the Alibaba taxonomy, shown in Table 3, displays a similar advantage for our method. In particular, the addition of instance data now gives an improvement even with the feed forward ensemble classifier. The baselines Onto2Vec, OPA2Vec and OWL2Vec* struggle more on this dataset since pre-trained word embeddings are not available in Chinese. Combined with Table 2, the results on the two datasets compared against four baselines strongly confirm the effectiveness of our model.

Ablation Studies. We now investigate the individual effects of the template and Tokenset. Table 4 presents the results on the eBay masked taxonomy recovery task, using the setting of Path+Instance with attention. A significant loss in all metrics when the template is removed indicates that setting an appropriate semantic context is very useful when representing oddly shaped textual data with BERT and that a suitable template is one way to achieve this. Tokenset can give an additional boost to performance when the template is present, but it barely improves efficacy without the template. This suggests that removing duplicate information from the input may help BERT identify and concentrate

Table 4. Results of different preprocessing settings on predicting masked subsumptions of the eBay taxonomy

Template	Tokenset	Feed forward				
		P	R	F1	MRR	H@5
no	no	.215	.786	.338	.422	.578
no	yes	.228	.760	.351	.419	.557
yes	no	.441	.817	.573	.529	.671
yes	yes	**.463**	**.835**	**.596**	**.552**	**.729**

on key segments related to the problem, but only when the appropriate context has already been established.

5.3 Observations

Complexities. Due to the downsampling and approximative nature of box embedding based prediction, it is able to save much time and space compared to k-NN prediction. For a task with m child instances and n parent instances, k-NN consumes $O(m \log n)$ time and $O(m + n)$ space, while box embedding consumes $O(m + n)$ time and $O(1)$ space. In reality, a single k-NN inference with $m = n = 1000$ takes around 0.1s on the hardware in Sect. 5.1, while box embedding takes a few milliseconds.

Dependency on Labels. The Tokenset process makes an important contribution to our model's competitive performance, but it also makes a hidden assumption on class labels. By converting the path to a flat keyword list, Tokenset essentially treats each label as a conjunction of constraints. This treatment is inappropriate when the label contains disjunctive parts, as exemplified in Sect. 4.2. While disjunction handling could be solved by a more sophisticated approach, there are taxonomies/ontologies where the labels are convoluted, technical phrases that any bag-of-words style treatment cannot tackle accurately, e.g., medical terminology ontologies. Therefore, our approach works well when class labels are relatively short and concise. Another reason to prefer short labels is that paths are more valuable in this case. Tokenset is effective in combining information from multiple labels, since its motivation is to address the scenario where the class's own label does not provide a full description, but its path does. This is the case in most e-commerce taxonomies. One can also apply Tokenset to label sequences other than paths, e.g., the breadth-first context corpus constructed in BERTSubs [5]. Overall, the Tokenset preprocessing helps clean noisy labels, resulting in a more compact contextual representation of the class.

Dependency on Instances. For instances to contribute to subsumption prediction, they must either have good quality, or have abundant quantity that compensates for the quality. This condition is naturally met in our case of e-commerce taxonomy with the large pool of items. However, the language models used in our

experiments are not pre-trained with e-commerce specific corpus, such as queries and item titles. It is a reasonable assumption that such pre-training can enhance the model's understanding of instance data and therefore overall performance. We will further investigate the role of instances and the impact of their quality in taxonomy curation in our future work.

6 Discussions and Conclusion

In this paper, we propose a new subsumption prediction model for taxonomies using PLMs such as BERT, with logical geometric embeddings and inferences. Inspired by the e-commerce setting, we design our model to utilise both class-level and instance-level information. At the class level, the model learns meaningful representations by using a template and preprocessing with lexical semantics to convert class labels into a concise list of tokens. At the instance level, we enrich the representations of class labels with instance data, and experiment with three classifier heads: feed forward, box embedding and extensional inference. Our evaluation on the eBay taxonomy in English and the Alibaba taxonomy in Chinese confirms our model's effectiveness. Furthermore, the experiments demonstrate the importance of templates and preprocessing, the advantage of instance-aware models with domain-specific PLM pretraining, and shows that box embedding is a promising alternative to the computationally expensive bruteforce method with either the feed forward classifier or direct extensional inference.

While this paper focuses on intra-taxonomy subsumption prediction, the techniques we describe could be applied to inter-taxonomy/ontology subsumption prediction. As shown in BERTSubs [5], inter-taxonomy subsumptions and indirect subsumptions inferred from existential restrictions can be expressed in templates and captured by PLMs. Furthermore, Tokenset and the modelling of ABox data can be directly generalised to the cases of inter-taxonomy and ontology.

Finally, we identify a few directions for future work: *Multi-modality* has drawn wide attention in recent machine learning and KG research [1,39]. E-commerce offers an ideal setting for investigating multi-modal KGs and multi-modal learning. Images, different kinds of properties and property values could be highly valuable complements to item titles for many taxonomy curation tasks, because most item titles are phrased to catch the eye and do not prioritise accurate and complete description of the item. Another interesting extension of this work would be taxonomy *enrichment* by inferring new classes from existing classes. In Fig. 1, we notice that applying facet constraints in different orders can lead to valid classes that are missing in the current taxonomy, e.g., *Vintage Clothing*, a concept that is currently split into *Men's Vintage Clothing* and *Women's Vintage Clothing* but does not exist on its own. The identification of these missing classes requires no external information, and in a sense fills the semantic "holes" of the taxonomy.

Acknowledgment. We would like to thank Mingjian Lu and Canran Xu for their work and constructive ideas. This work was supported by eBay and the EPSRC projects OASIS (EP/S032347/1), UK FIRES (EP/S019111/1) and ConCur (EP/V050869/1).

References

1. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423–443 (2018)
2. Bechhofer, S.: OWL web ontology language reference, W3C recommendation (2004). http://www.w3.org/TR/owl-ref/
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
4. Centelles, M.: Taxonomies for categorization and organization in web sites. Hipertext.net (3) (2005). https://www.upf.edu/hipertextnet/en/numero-3/taxonomias.html
5. Chen, J., He, Y., Jimenez-Ruiz, E., Dong, H., Horrocks, I.: Contextual semantic embeddings for ontology subsumption prediction. arXiv preprint arXiv:2202.09791 (2022)
6. Chen, J., Hu, P., Jimenez-Ruiz, E., Holter, O.M., Antonyrajah, D., Horrocks, I.: OWL2Vec*: embedding of owl ontologies. Mach. Learn. **110**(7), 1813–1845 (2021)
7. Cruse, D.A.: Hyponymy and Its Varieties. In: Green, R., Bean, C.A., Myaeng, S.H. (eds.) The Semantics of Relationships, pp. 3–21. Springer, Dordrecht (2002). https://doi.org/10.1007/978-94-017-0073-3_1
8. Dasgupta, S., Boratko, M., Zhang, D., Vilnis, L., Li, X., McCallum, A.: Improving local identifiability in probabilistic box embeddings. Adv. Neural. Inf. Process. Syst. **33**, 182–192 (2020)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dhingra, B., Shallue, C.J., Norouzi, M., Dai, A.M., Dahl, G.E.: Embedding text in hyperbolic spaces. arXiv preprint arXiv:1806.04313 (2018)
11. Dong, H., Wang, W., Coenen, F.: Rules for inducing hierarchies from social tagging data. In: Chowdhury, G., McLeod, J., Gillet, V., Willett, P. (eds.) iConference 2018. LNCS, vol. 10766, pp. 345–355. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78105-1_38
12. Gumbel, E.J.: Les valeurs extrêmes des distributions statistiques. In: Annales de l'institut Henri Poincaré, vol. 5, pp. 115–158 (1935)
13. Iqbal, R., Murad, M.A.A., Mustapha, A., Sharef, N.M., et al.: An analysis of ontology engineering methodologies: a literature review. Res. J. Appl. Sci. Eng. Technol. **6**(16), 2993–3000 (2013)
14. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2019)
15. Jurgens, D., Pilehvar, M.T.: Semeval-2016 task 14: semantic taxonomy enrichment. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 1092–1102 (2016)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. arXiv preprint arXiv:2205.11916 (2022)

18. Lee, S., Park, Y.: The classification and strategic management of services in e-commerce: development of service taxonomy based on customer perception. Expert Syst. Appl. **36**(6), 9618–9624 (2009)
19. Lees, A., Welty, C., Zhao, S., Korycki, J., Mc Carthy, S.: Embedding semantic taxonomies. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1279–1291 (2020)
20. Li, X., Vilnis, L., Zhang, D., Boratko, M., McCallum, A.: Smoothing the geometry of probabilistic box embeddings. In: International Conference on Learning Representations (2018)
21. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
22. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786 (2021)
23. Luo, X., et al.: Alicoco: alibaba e-commerce cognitive concept net. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 313–327 (2020)
24. Mao, Y., et al.: Octet: online catalog taxonomy enrichment with self-supervision. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2247–2257 (2020)
25. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
26. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
27. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
28. Nickel, M., Kiela, D.: Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In: International Conference on Machine Learning, pp. 3779–3788. PMLR (2018)
29. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. Semant. Web **8**(3), 489–508 (2017)
30. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Discov. Data (TKDD) **15**(2), 1–49 (2021)
31. Schafer, J.B., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. Data Min. Knowl. Disc. **5**(1), 115–153 (2001)
32. Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5149–5152. IEEE (2012)
33. Smaili, F.Z., Gao, X., Hoehndorf, R.: Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. Bioinformatics **34**(13), i52–i60 (2018)
34. Smaili, F.Z., Gao, X., Hoehndorf, R.: OPA2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics **35**(12), 2133–2140 (2019)
35. Stock, W.G.: Concepts and semantic relations in information science. J. Am. Soc. Inform. Sci. Technol. **61**(10), 1951–1969 (2010)
36. Vendrov, I., Kiros, R., Fidler, S., Urtasun, R.: Order-embeddings of images and language. arXiv preprint arXiv:1511.06361 (2015)
37. Vilnis, L., Li, X., Murty, S., McCallum, A.: Probabilistic embedding of knowledge graphs with box lattice measures. arXiv preprint arXiv:1805.06627 (2018)

38. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
39. Zhu, X., et al.: Multi-modal knowledge graph construction and application: a survey. arXiv preprint arXiv:2202.05786 (2022)