Representation Learning for Hierarchical Classification of Entity Titles

Elena Chistova
FRC CSC RAS, Moscow, Russia
chistova@isa.ru

Abstract

We present a method for effective title encoding for hierarchical classification in a large taxonomy. The method enables taxonomy-aware encoding in pre-trained text encoders, such as fastText and BERT, which are additionally fine-tuned for the hierarchical classification. The embeddings produced using our method perform well when applied to nearest neighbor classification. They allow for controllable and sufficient hierarchical classification based solely on the title.

1 Introduction

Hierarchical classification is the task of organizing data into a hierarchy of categories, where each category is a subset of another category. This structure can be thought of as a tree-like structure, where the root node represents the most general category and the leaf nodes represent the most specific categories. In NLP, hierarchical text classification (HTC) is widely used to organize large collections of documents (e.g. emails, patents, job advertisements, digital libraries) or entities (e.g. product or service titles in e-commerce). This work focuses on the challenge of inferring fine-grained categories from no other information but an entity name, which is a specific challenge for hierarchical classification.

The deep hierarchical classification approaches developed over the past years (Yang et al., 2020; Gao, 2020; Gong et al., 2023) have three major limitations:

- Entity HTC models are often developed for e-commerce and use multiple attributes for the input entity including detailed descriptions, tags, or images. However, there are other situations where just the textual titles are available for classification, like mapping diagnoses and procedures to a clinical coding taxonomy (Li et al., 2019; Chakraborty et al., 2023). Better title representations can also be beneficial when multiple attributes are present.
- Being mostly deep learning classification methods, they are prone to class imbalance and may not be able to handle large skewed hierarchies with a few examples per leaf.
- Limited interpretation capabilities of the deep hierarchical classifiers are another disadvantage that can be critical in some practical applications.

To address these limitations, we propose a simple yet effective approach that encodes the textual title using hierarchy-aware information to map an object's title to the relevant leaf in the taxonomy. We show that our approach improves the classification performance of deep models while making the entity title classification easier to interpret and control[1].

2 Related Work

Hierarchical Entity Title Classification In hierarchical classification, each object is associated with a certain branch (labels path) in the hierarchy tree. There are three fundamental approaches to hierarchical classification: flat classification (object-to-branch), global classification, and local classification (Silla and Freitas, 2011). Global classification predicts classes in the hierarchy using a single model that considers class dependencies, whereas local classification uses multiple separate models for different hierarchy nodes or levels.

Previous approaches to HTC for e-commerce mainly focus on title-plus-description classification, and include flat classifiers (Skinner, 2018; Suzuki et al., 2018), two-level pipelines (Cevahir and Murakami, 2016; Gupta et al., 2016; Das et al., 2017; Goumy and Mejri, 2018), multilabel classifiers (Jia et al., 2018; Yu et al., 2018), and sequence-to-sequence branch generation (Li et al., 2018).

[1]The code is available at https://github.com/tchewik/entity_representation_learning

Shared tasks often feature the systems investigating external ways to improve classification performance, including model ensembling (Yang et al., 2020; Yu et al., 2018; Jia et al., 2018), pseudo labeling (Yang et al., 2020), and collecting additional data (Borst et al., 2020). Some approaches focus on optimizing the classification model itself by considering the hierarchy of classes in the activation (Yang et al., 2020) or loss (Gao, 2020) function. Other methods involve matching an entity title with a leaf title (Chen et al., 2021; Gong et al., 2023). To improve entity title encoding for product classification and overcome the problem of domain shift, Brinkmann and Bizer (2021) suggest additionally pre-training the transformer on product offers from Common Crawls.

In our method, we train a single global deep classifier and utilize it to encode entity titles in a complicated hierarchy for flat categorization. We demonstrate that this approach excels in terms of accuracy on the deepest levels of hierarchy, simplicity, and controllability.

LLM Applications Large language models have limited structured prediction capabilities. There have been recent attempts to solve the HTC task through hierarchy verbalization, however, they still rely on pretrained BERT rather than LLMs and require model architecture modifications: Wang et al. (2022) frame the problem as a hierarchy-aware multi-label MLM task, adopting a Graph Attention Network and a zero-bounded Multi-label Cross-Entropy Loss, while Ji et al. (2023) address HTC as flat classification solvable by verbalizing with a hierarchy-aware decoder constraint. Although promising, these methods are tailored and evaluated for elaborate texts in smaller taxonomies (WOS, DBPedia, RCV1-V2).

While prompting LLMs for this task can be possible for flat entity title classification in a large hierarchy, there are some major limitations:

- A large language model should memorize an entire deep taxonomy with thousands of branches and adhere to its complex structure without deviation. This level of precision is achievable by imposing low-level constraints overriding the NLG capabilities of LLMs. Constraining LLMs in this way erases their main strength in favor of precise taxonomic compliance – an outcome more efficiently reached by fine-tuning text encoders.
- Few-shot learning is successful in many tasks, but it is not suitable for the hierarchical classification in a large taxonomy. Exposing the LLM to examples spanning all the taxonomy branches, or fine-tuning on a large labeled dataset, would be extremely resource- and time-intensive.
- LLM predictions cannot be controlled or interpreted precisely. This lack of transparency makes LLMs unsuitable for settings requiring controllable accuracy and recall.

3 Background

In this work, we compare nearest-neighbors classification, deep hierarchical classification, and our hybrid method as three basic approaches to entity title classification in a large taxonomy.

3.1 k-Nearest-Neighbor Classification

Given representations of entity titles in hierarchically organized data, the embedding of an input entity is assigned to a leaf of the hierarchy based on the leaves of its k nearest neighbors. The distance between text embeddings is typically estimated as a cosine distance, and k nearest neighbor classes are weighted according to the distances.

Advantages: (1) The most interpretable method. (2) With small k is immune to subclass imbalance in a complex hierarchy.

Disadvantages: (1) Domain shift affects pretrained language models substantially, and domain adaptation requires additional resources for data collection and computation. (2) With small k, highly sensitive to outliers. (3) Does not provide any information about the taxonomy.

3.2 Deep Hierarchical Classification

The classifier predicts the most probable classes for each level of the hierarchy and collects the final prediction from a pool of weighted class labels. The classifier can predict multiple labels in a multi-label fashion or have n top outputs for all hierarchy levels.

Advantages: (1) The internal representations of texts in the neural model are influenced by both their own surface forms and their position in the hierarchy. (2) More robust to data noise. (3) Can more or less adjust to specific domains while fine-tuning.

Disadvantages: (1) Is highly affected by class imbalance. (2) Has reduced interpretability. (3) As

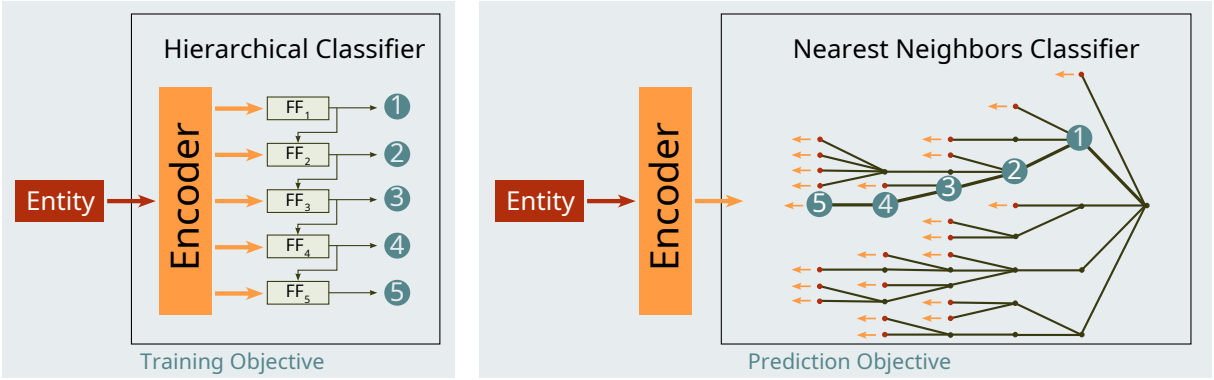

Figure 1: Overview of our framework with DBERT$_{1-5}$ as a deep classifier. During training, the encoder is paired with outputs for hierarchical classification. The classification part of the model is fine-tuned in conjunction with the encoder to generate a sequence of subclass labels (levels 1-5). During inference, we encode the known data and input entity using only the fine-tuned encoder and attempt to find the most similar entities in a complex taxonomy. Finally, we assign the input entity to the hierarchy leaf with the most similar known entities.

a result of the previous point, it is more difficult to control the precision of the model when implementing it in real-world systems. Classifier confidence is not transparent. (3) The model itself can produce contradictory labels (non-existing taxonomy branches), and introducing hierarchical information can require the implementation of additional restrictions.

4 Methods

We compare multiple methods that follow the two fundamental strategies introduced in Section 3. The described HTC methods employing a single model (FT$_{1-5}$, DBERT$_{1-5}$) are additionally probed in the hybrid classification setting.

4.1 k-NN

The most similar titles in a hierarchy are found using cosine distance. The out-of-the-box encoder is not fine-tuned on task-related data. The title is encoded as an average of token representations. We probe two types of representations: **fastText** and **DeBERTa**.

4.2 Trainable Classification

A deep classification model simultaneously predicts multiple labels denoting the nodes in a hierarchy. The final prediction assigns an entity title to a taxonomy branch and is constructed from top-n predicted node labels along with their probabilities.

FT$_{1-5}$: To predict top-n possible nodes for levels 1-5, we use a one-vs-all multilabel classification implemented in the fastText[2] library.

[2] https://fasttext.cc/

DBERT$_{1-5}$: We use an architecture of a deep hierarchical classifier similar to that of Gao (2020). The output layers for every level are added on top of an encoding language model (DeBERTa). For the title consisting of tokens $w_1, w_2, ..., w_z$, the representations are computed in encoder:

$$e = \text{Encoder}(w_1 w_2 ... w_k) \in \mathbb{R}^{d_{LM}} \quad (1)$$

The output for each hierarchy level i is predicted with a separate feedforward layer. Input for the output layer $i > 1$ is a concatenation of the text embedding e and an output for the previous level:

$$y_i = \begin{cases} \text{FF}_i(e) & \text{if } i = 1; \\ \text{FF}_i(e \oplus y_{i-1}), & \text{otherwise.} \end{cases} \quad (2)$$

The probabilities of classes for a hierarchy level i are calculated by passing y_i through the softmax activation function. The class with the highest predicted probability is then predicted as $\hat{y}_i$. The loss function is a weighted sum of the categorical cross-entropy loss and hierarchical loss:

$$\text{HLoss}_i = \begin{cases} 0 & \text{if } \hat{y}_i \subset \hat{y}_{i-1}; \\ 1 & \text{otherwise.} \end{cases}$$
$$\text{Loss} = \alpha \sum_{i=1}^{n} \text{CELoss}_i + \sum_{i=2}^{n} \beta^{i-1} \text{HLoss}_i \quad (3)$$

where α and β are the weights controlling the impact of hierarchical loss. The hyperparameter β ($0 < \beta \leq 1$) is used to scale the hierarchical loss. The cross-entropy loss is weighted to handle the class imbalance on each hierarchy level.

Part	Deduplicated Length	Unique Branches	Unique Classes of Each Level 1	2	3	4	5
Clothing Shoes and Jewelry	1988301	35710	11	253	953	5263	12371
Home and Kitchen	1203754	1671	13	136	539	695	292
Automotive	831549	2252	14	165	743	849	318
Sports and Outdoors	809999	3414	3	43	351	1102	1281
Electronics	584136	900	16	105	290	305	133
Tools and Home Improvement	488042	1152	13	98	435	427	172
Industrial and Scientific	132168	1796	25	301	970	496	93

Table 1: Statistics of the corpus.

4.3 Our hybrid approach

As a compromise between both of the described methods, we propose a hybrid approach, in which the out-of-the-box text encoder is additionally pretrained on hierarchical classification. The overall framework is illustrated in Figure 1. The title encoder is fine-tuned as a part of a hierarchical classifier, and the nearest-neighbor classifier dealing with flat (entity; leaf) pairs predicts the leaf with most similar entities.

5 Experimental Setup

5.1 Dataset

Deep models with millions of parameters, such as BERT, have a tendency to overfit to noise and outliers in e-commerce product classification data, as noted by Zhang et al. (2021). They describe two major challenges in e-commerce data: frequently incomplete or misleading item descriptions and confusing or non mutually exclusive labels in a large taxonomy. Supervised learning faces a significant obstacle when classifying images, descriptions, or titles due to confusing and non-mutually exclusive labels in a large taxonomy. To address this issue, we thoroughly clean the data for our experiments.

We only use the titles and hierarchy annotations from the Amazon review dataset[3] (Ni et al., 2019); HTML character references in both titles and categories are decoded into Unicode. We cut subbranches leaving only the nodes containing less than 13 tokens[4] in name and keep only subbranches appearing in the data at least 4 times. We have selected seven major data subsets that have at least 90 classes annotated in the 5th level of the hierarchy. The statistics of the obtained data are described in Table 1. On each hierarchy level, we encode classes independently of the previous levels. As a result, on most subsets, the number of classes decreases after level 4; instead, "missing" class replacement occurs most frequently. This denotes a natural skew in the hierarchy.

5.2 Metrics

We evaluate the hierarchical classification performance with 5-fold stratified cross-validation. This balances the distribution of branches in each fold. Firstly, we calculate macro-averaged F1 for each level of the hierarchy. Since this F1 reflects performance for each level independently, we also evaluate the accuracy for flat branch assignment for each depth.

5.3 Implementation Details

fastText We use a fastText model described in (Grave et al., 2018) that is pretrained on Common Crawl and Wikipedia data. *Hierarchical model (FT_{1-5}):* The classifier is fine-tuned using the one-vs-all scheme, with a learning rate of 1, character n-gram range of (3, 10), and for 25 epochs. The top 7 predicted nodes are used to assemble the full branch after classification.

Contextual Embeddings As a pretrained transformer, we employ DeBERTa[5] (He et al., 2021). *Hierarchical model (DBERT_{1-5}):* The model is fine-tuned with a learning rate of 2e-5, dropout rate of 0.4, batch size of 128, $\alpha = 1$, $\beta = 0.9$, and the cross-entropy loss for each level (CELoss_i in (3)) is weighted based on the distribution of classes in the subcorpus. The top 8 predicted nodes are used to assemble the final branch.

[3] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/

[4] We considered the longer nodes noisy because they often included non-taxonomy information, such as notes for customers (e.g. "*Please feel free to contact us if you have any special requests or questions*") or lengthy keyword-stuffed descriptions (e.g. "*My Daily Styles Stainless Steel Black Faux PU Leather Yellow Gold-Tone Latin Cross Religious Adjustable Wristband Mens Bracelet*") hardly resembling subclasses.

[5] microsoft/deberta_v3_base

6 Experimental Results

Table 2 compares all the investigated methods for hierarchical classification. The statistics of macro F1 calculated for each level independently are illustrated in Figure 2.

6.1 Baselines

The results for kNN using out-of-the-box pretrained text encoders are denoted as KNN:FT for fastText and KNN:DBERT for DeBERTa. The fastText-based flat kNN classifier provides a strong baseline across all subcorpora. The low performance of the KNN:DBERT can be attributed to a known issue with transformers: the feature extraction performance of the frozen model decreases with increasing difference between pretraining and target tasks (Peters et al., 2019).

6.2 Trainable Classifiers

The fastText- and DeBERTa-based classifiers are denoted as FT_{1-5} and DBERT_{1-5}, respectively.

According to the results in Table 2, the fastText-based hierarchical classifier outperforms the kNN baseline only across the smallest subcorpora, and mostly for the higher levels of hierarchy. Moreover, for larger datasets, starting with "Sports and Outdoors" multilabel fastText training becomes increasingly more challenging and consuming. The statistics of hierarchical labels are actually learned by the model, which we'll see by applying kNN to its representations. However, collecting the taxonomy branch from top-n pool of predicted labels using the direct approach is hardly applicable.

DBERT_{1-5} outperforms not only the corresponding weak baseline but also the fasttext-based hybrid classification KNN:FT_{1-5} on many datasets. It is also worth noting that this method handles larger data with larger class sets much better than multilabel fastText.

6.3 k-NN over the Tuned Representations

Applying kNN directly to the inner representations results in an improvement in classification for all levels for both backbones (KNN:FT_{1-5} and KNN:DBERT_{1-5}). In addition to a considerable improvement in the accuracy of full branch prediction (A_{1-5} in Table 2) while preserving or improving the intra-level F1 (Figure 2), the purely vector-based approach can also be significantly faster than collecting known branches from a pool of predicted labels for each entity.

	A_1	A_{1-2}	A_{1-3}	A_{1-4}	A_{1-5}
	Clothing Shoes and Jewelry				
KNN:FT	85.5	81.4	71.0	55.3	44.4
FT_{1-5}	84.1	76.8	64.1	45.4	31.1
KNN:FT_{1-5}	86.8	83.1	73.3	57.8	45.9
KNN:DBERT	72.7	64.3	51.3	38.9	31.8
DBERT_{1-5}	90.8	88.2	80.3	65.7	52.7
KNN:DBERT_{1-5}	**90.9**	**88.4**	**80.8**	**67.0**	**54.9**
	Home and Kitchen				
KNN:FT	89.7	78.5	68.5	64.4	63.5
FT_{1-5}	91.5	80.0	68.0	62.9	60.6
KNN:FT_{1-5}	90.9	81.0	71.5	67.4	66.5
KNN:DBERT	64.5	49.6	42.2	39.8	39.8
DBERT_{1-5}	93.3	85.0	76.5	72.7	71.6
KNN:DBERT_{1-5}	**93.6**	**85.6**	**77.6**	**74.0**	**73.1**
	Automotive				
KNN:FT	89.4	82.1	76.1	72.7	72.0
FT_{1-5}	88.5	80.1	72.9	67.9	66.6
KNN:FT_{1-5}	91.8	86.0	80.7	77.4	76.7
KNN:DBERT	76.9	66.9	61.3	58.7	58.3
DBERT_{1-5}	92.1	86.3	80.8	77.4	76.6
KNN:DBERT_{1-5}	**92.3**	**86.9**	**81.8**	**78.6**	**77.8**
	Sports and Outdoors				
KNN:FT	91.8	81.7	73.0	64.2	59.3
FT_{1-5}	90.3	78.0	67.1	56.7	50.3
KNN:FT_{1-5}	93.3	85.2	77.5	69.2	64.6
KNN:DBERT	77.0	54.5	46.0	40.8	38.0
DBERT_{1-5}	94.4	87.3	80.3	72.6	68.0
KNN:DBERT_{1-5}	**94.5**	**87.8**	**81.2**	**74.0**	**69.8**
	Electronics				
KNN:FT	87.0	76.3	68.6	64.0	62.6
FT_{1-5}	87.4	74.6	64.8	58.3	56.7
KNN:FT_{1-5}	89.4	79.8	72.6	68.5	67.2
KNN:DBERT	63.9	50.3	43.6	40.4	39.6
DBERT_{1-5}	89.8	80.1	72.5	68.1	66.9
KNN:DBERT_{1-5}	**90.1**	**80.8**	**73.7**	**69.5**	**68.3**
	Tools and Home Improvement				
KNN:FT	88.3	78.9	68.4	64.3	62.9
FT_{1-5}	89.9	79.8	69.2	63.7	62.1
KNN:FT_{1-5}	91.9	84.3	75.2	70.9	69.6
KNN:DBERT	62.0	51.7	43.9	41.7	40.8
DBERT_{1-5}	92.2	84.6	75.6	71.1	69.8
KNN:DBERT_{1-5}	**92.4**	**85.2**	**76.5**	**72.2**	**70.9**
	Industrial and Scientific				
KNN:FT	82.0	71.8	63.6	60.6	60.2
FT_{1-5}	85.1	74.1	64.7	60.3	59.7
KNN:FT_{1-5}	87.9	79.1	71.4	68.2	67.8
KNN:DBERT	56.8	49.0	44.1	42.4	42.3
DBERT_{1-5}	88.2	78.9	70.6	67.4	67.0
KNN:DBERT_{1-5}	**88.4**	**79.4**	**71.5**	**68.5**	**68.0**

Table 2: Mean accuracy of the branch prediction. The datasets are listed in descending order of size (see Table 1).

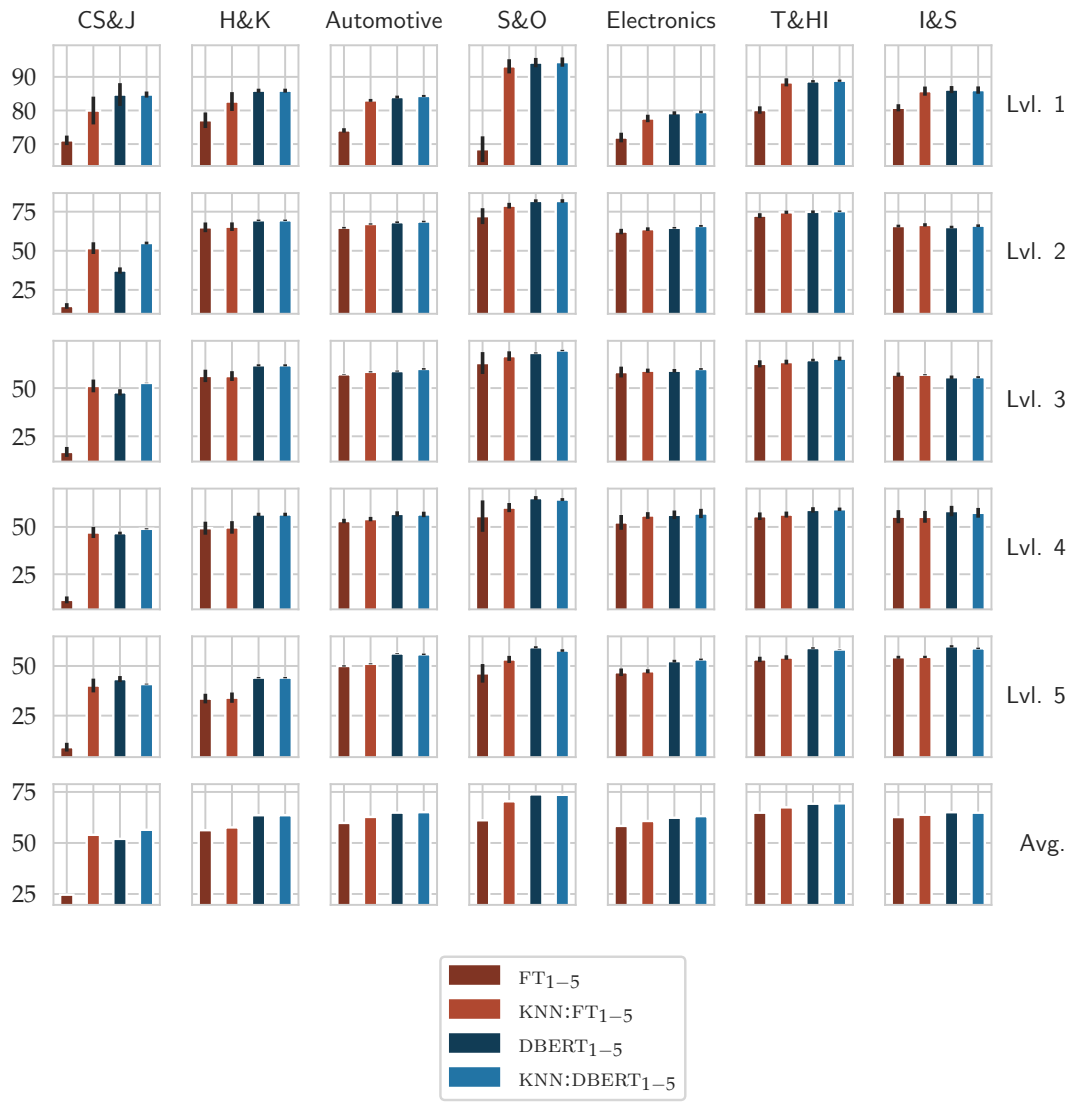

Figure 2: Macro F1 calculated for each level independently; two types of classifiers.

7 Conclusion

We present an approach for entity title hierarchical classification that uses representation learning for training hierarchy-informed embeddings. We apply the obtained embeddings in kNN flat hierarchical classification to demonstrate how these representations can be directly used in a controllable setting. The baselines include pretrained encoders used as the base encoders in the pipeline and hierarchical classifiers built with the same encoders. The hybrid approach outperforms the baselines on each part of the large-taxonomy e-commerce corpus.

Acknowledgments

The research was carried out using the infrastructure of the Shared Research Facilities «High Performance Computing and Big Data» (CKP «Informatics») of FRC CSC RAS (Moscow).

References

Janos Borst, Erik Korner, Kobkaew Opasjumruskit, and Andreas Niekler. 2020. Language model CNN-driven similarity matching and classification for HTML-embedded product data. In *Proceedings of the semantic web challenge on mining the web of HTML-embedded product data co-located with the 19th international semantic web conference*.

Alexander Brinkmann and Christian Bizer. 2021. Improving hierarchical product classification using domain-specific language modelling. In *Proceedings of Workshop on Knowledge Management in e-Commerce @ The Web Conference '21*, volume 44, pages 14–25.

Ali Cevahir and Koji Murakami. 2016. Large-scale multi-class and hierarchical product categorization

for an E-commerce giant. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 525–535, Osaka, Japan. The COLING 2016 Organizing Committee.

Sinchani Chakraborty, Harsh Raj, Srishti Gureja, Tanmay Jain, Atif Hassan, and Sayantan Basu. 2023. Evaluating the robustness of biomedical concept normalization. In *Proceedings of The 1st Transfer Learning for Natural Language Processing Workshop*, pages 63–73. PMLR.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware label semantics matching network for hierarchical text classification. 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 4370–4379, Online. Association for Computational Linguistics.

Pradipto Das, Yandi Xia, Aaron Levine, Giuseppe Di Fabbrizio, and Ankur Datta. 2017. Web-scale language-independent cataloging of noisy product listings for E-commerce. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 969–979, Valencia, Spain. Association for Computational Linguistics.

Dehong Gao. 2020. Deep hierarchical classification for category prediction in E-commerce system. In *Proceedings of the 3rd Workshop on e-Commerce and NLP*, pages 64–68, Seattle, WA, USA. Association for Computational Linguistics.

Shansan Gong, Zelin Zhou, Shuo Wang, Fengjiao Chen, Xiujie Song, Xuezhi Cao, Yunsen Xian, and Kenny Zhu. 2023. Transferable and efficient: Unifying dynamic multi-domain product categorization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 476–486, Toronto, Canada. Association for Computational Linguistics.

Sylvain Goumy and Mohamed-Amine Mejri. 2018. Ecommerce product title classification. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Vivek Gupta, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. 2016. Product classification in E-commerce using distributional semantics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 536–546, Osaka, Japan. The COLING 2016 Organizing Committee.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing.

Ke Ji, Yixin Lian, Jingsheng Gao, and Baoyuan Wang. 2023. Hierarchical verbalizer for few-shot hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2918–2933, Toronto, Canada. Association for Computational Linguistics.

Yugang Jia, Xin Wang, Hanqing Cao, Boshu Ru, and Tianzhong Yang. 2018. An empirical study of using an ensemble model in e-commerce taxonomy classification challenge. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Fei Li, Yonghao Jin, Weisong Liu, Bhanu Pratap Singh Rawat, Pengshan Cai, Hong Yu, et al. 2019. Fine-tuning bidirectional encoder representations from transformers (bert)–based models on large-scale electronic health record notes: an empirical study. *JMIR medical informatics*, 7(3).

Maggie Yundi Li, Liling Tan, Stanley Kok, and Ewa Szymanska. 2018. Unconstrained product categorization with sequence-to-sequence models. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72.

Michael Skinner. 2018. Product categorization with LSTMs and balanced pooling views. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Shogo D. Suzuki, Yohei Iseki, Hiroaki Shiino, Hongwei Zhang, Aya Iwamoto, and Fumihiko Takahashi. 2018. Convolutional neural network and bidirectional lstm based taxonomy classification using external dataset at sigir ecom data challenge. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Li Yang, E Shijia, Xu Shiyao, and Xiang Yang. 2020. Bert with dynamic masked softmax and pseudo labeling for hierarchical product classification. In *Proceedings of the semantic web challenge on mining the web of HTML-embedded product data co-located with the 19th international semantic web conference*.

Wenhu Yu, Zhiqiang Sun, Haifeng Liu, Zhipeng Li, and Zhitong Zheng. 2018. Multi-level deep learning based e-commerce product categorization. In *Proceedings of the Workshop on eCommerce (co-located with SIGIR)*.

Wen Zhang, Yanbin Lu, Bella Dubrov, Zhi Xu, Shang Shang, and Emilio Maldonado. 2021. Deep hierarchical product classification based on pre-trained multilingual knowledge. *IEEE Data Eng. Bull.*, 44(2):26–37.