DNA-HINT: Domain-novelty Aware Hierarchical Introspection for Hierarchical Novelty Detection

Anonymous ACL submission

Abstract

Deep neural networks have achieved impressive performance for text classification that recognizes a predefined set of classes. However, recognizing texts of novel classes unseen during training is not well explored. It is desirable for large-scale text datasets to augment a function of detecting the novelty of a newly-joined text, especially in practical application scenarios such as an e-commerce system. We aim to achieve a hierarchical novelty detection that predicts the closest known class in the taxonomy for a text of a novel class. Furthermore, existing approaches typically encounter issues, such as (i) the inconsistency problem that the predictions in any pair of parent-child nodes are not successive; (ii) the blocking problem that the prediction at a certain level is not confident and unable to be passed downward in the taxonomy; (iii) the overconfidence problem of a softmax classifier that predicts high confidence regardless of whether a text is a known or novel class. In this paper, we propose a novel model, Domain-Novelty Aware Hierarchical Introspection (DNA-HINT), to achieve the goal without those problematic issues. Extensive experiments conducted on four benchmark datasets show that DNA-HINT is effective particularly for deep levels that are often considered in realistic scenarios.

1 Introduction

Text classification has achieved impressive performance with the transformer model (Devlin et al., 2019) to recognize a predefined set of classes. However, large-scale text datasets in practical application scenarios such as an e-commerce system or an Internet-based encyclopedia often have a naturally hierarchical structure and encounter newly-joined texts from time to time. Thus, it is desirable to augment a function of detecting the novelty of a text with a hierarchical taxonomy (i.e., differentiating whether a text conforms to any previously trained classes and categorizing it to the closest known class if predicted as a novel class). For example in Figure 1, we aim to achieve a hierarchical novelty detection task (Lee et al., 2018a) that predicts with more fine-grained labels, such as "Novel Electronics for Kids", "Novel Games", and "Novel Toys Games".

We are also motivated by the challenges of hierarchical classification and novelty detection. The top-down method is widely explored in the literature of hierarchical classification, which takes the advantage of structural and local information. Specifically, it often has a set of local classifiers that make predictions in a top-down manner. There are two major drawbacks of it. One is the inconsistency problem that the predictions in any pair of parent-child nodes are not successive, and the other is the blocking problem that the prediction at a certain level is not confident and unable to be passed downward (Sun and Lim, 2001; Mao et al., 2019; Gao et al., 2020). Furthermore, a softmax classifier that is commonly used for novelty detection (Hendrycks and Gimpel, 2017) suffers from the overconfidence problem, i.e., predicting high confidence regardless of whether a text is a known or novel class (Lakshminarayanan et al., 2016; Guo et al., 2017).

To address these problems, in this paper, we propose a novel model, **D**omain-**N**ovelty **A**ware **H**ierarchical **Int**rospection (**DNA-HINT**), that can differentiate whether a newly-joined text conforms to any previously trained classes on the taxonomy built with known classes and categorize it to the closest known class if predicted as a novelty. DNA-HINT consists of three components: a semantic encoder, a domain-novelty aware network, and a hierarchical introspection network.

For evaluation, we propose a novel metric to answer the inadequacy of existing metrics. Extensive experiments show that DNA-HINT significantly outperforms the baseline on four benchmark datasets: Amazon, DBPedia, 20 Newsgroups, and

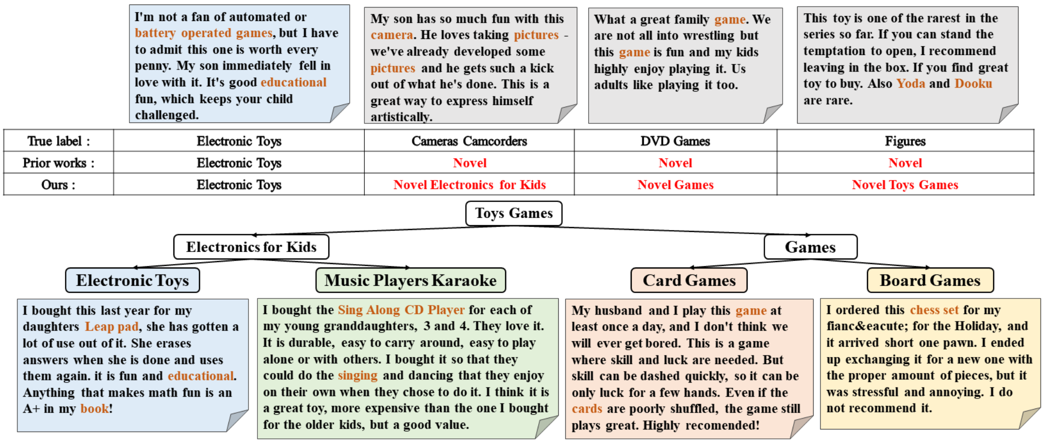

Figure 1: An illustration of our hierarchical novelty detection task in the Amazon dataset. The brown words are remarked for mentioning the name of the product.

Reuters 52.

The contributions of this paper are as follows:

- We propose a novel domain-novelty aware hierarchical introspection model for hierarchical novelty detection that can distinguish text into finer-grained known and novel classes. The integrated framework of DNA-HINT naturally solved the blocking problem
- The domain-novelty aware network can explicitly consider the effect of domain to avoid overconfident prediction.
- The hierarchical introspection network can estimate the inconsistency errors hierarchically and accordingly compute the loss.
- Our proposed measure can give adequate credit with respect to the correctness for both the classification of each level and the identification of novelty.
- On four benchmark datasets, DNA-HINT significantly outperforms the baseline and is particularly effective for the lowest-level detection that is most important in practical applications.

2 Related Work

2.1 Hierarchical Classification

Hierarchical classification approaches are to address a classification problem with a pre-established class taxonomy, which is often a tree-structured hierarchy that any parent-child relationship satisfies the four properties (Wu et al., 2005). The approaches usually vary by the traversal method of the structure (Freitas and de Carvalho, 2007; Sun and Lim, 2001), which is categorized as the top-down (or local) method, i.e., a set of local classifiers that make predictions in a top-down manner, the global method, i.e., a single classifier manages the prediction of the entire hierarchy (Qiu et al., 2009), and the flat method, i.e., classifiers predict the leaf nodes only (Johnson and Zhang, 2014). Many previous studies train a set of multi-class classifiers that operate independently, which may suffer from the blocking and inconsistency problems during inference (Sun and Lim, 2001; Mao et al., 2019; Gao et al., 2020).

Selecting appropriate evaluation metrics is also an important issue. Most researchers used standard flat classification evaluation metrics, such as accuracy, precision, and recall, while recognizing that they are not ideal because errors at different levels are not considered (Silla and Freitas, 2011). The hierarchical metrics of precision (hP), recall (hR), and f-measure (hF_1) are proposed by Kiritchenko and Famili (2005) for evaluating hierarchical classification approaches, where correct predictions in different heights are differentially considered.

hF_1 is computed by calculating hP and hR for each input x_i with targeted label y_i and predicted label $\tilde{y}$:

$$hP = \frac{\sum_i |A(y) \cap A(\tilde{y})|}{\sum_i |A(\tilde{y})|} \quad hR = \frac{\sum_i |A(y) \cap A(\tilde{y})|}{\sum_i |A(y)|}$$

where $A(y)$ and $A(\tilde{y})$ denote the set of ancestor classes for y and $\tilde{y}$, respectively. Then, hF_1 is defined as:

$$hF_1 = \frac{2 \cdot hP \cdot hR}{hP + hR}$$

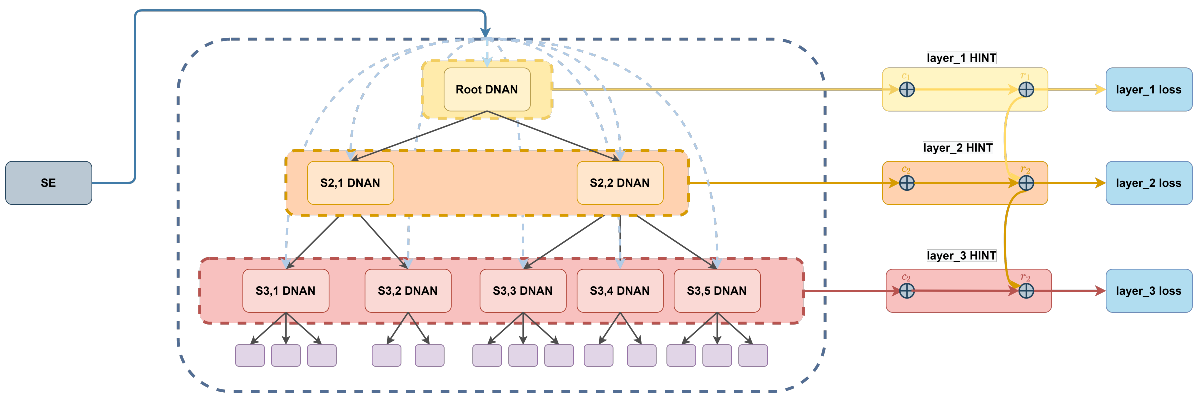

Figure 2: An illustration of our proposed Domain-novelty Aware Hierarchical Introspection model (DNA-HINT). SE denotes the semantic encoder, DNAN denotes the domain-novelty aware network, and HINT denotes the hierarchical introspection network.

2.2 Novelty Detection

Novelty detection is the identification of novel instances that are significantly different from the representative training data, which is often called novelty detection, outlier detection, or out-of-distribution detection (Hodge and Austin, 2004; Hendrycks et al., 2020). Many studies put efforts on threshold-based classifiers that compare the confidence score to some threshold $\delta > 0$ and commonly evaluate the performance with the AUROC (area under the receiver operating characteristic curve), which discriminates all possible thresholds (Lee et al., 2018b; Li et al., 2021). Hendrycks and Gimpel (2017) defined the maximum softmax probability (MSP) as the confidence score and presented the MSP as a baseline model of novelty detection in various domains, including computer vision (CV), automatic speech recognition (ASR), and natural language processing (NLP). Hsu et al. (2020) proposed a decomposed confidence method to address the overconfidence problem of a softmax classifier (Lakshminarayanan et al., 2016; Guo et al., 2017) by explicitly taking the influence of domain into consideration. That is, instead of predicting $p(y, x)$, a classifier using the decomposed confidence is defined as:

$$p(y|d_{in}, x) = \frac{p(y, d_{in}|x)}{p(d_{in}|x)}$$

where $p(y|d_{in}, x)$ is the decomposed confidence and d_{in} is a binary domain variable indicating whether a text belongs to any known class or not in the decomposed conditional probability.

3 Task Definition

Let $D^{train} = \{x, y^{train}\}$ and $D^{test} = \{x, y^{test}\}$ be two sets independently used for model training and test, where x denote the texts, the labels for training $y^{train} = \{1, ..., k\}$ consists of k distinct known labels, and the labels for test $y^{test} = \{1, ..., k, k+1, ..., k+t\}$ consists of the labels in D^{train} plus t additional novel labels. We assume a discriminative model is trained on D^{train}, and tested on D^{test}.

Let $T = (S, E)$ be a taxonomy with L levels. S is a set of nodes (classes) consisting of the known and novel class labels in y^{train} and y^{test} as external nodes and their ancestors including the root as internal nodes, which s denotes any internal node and s_{li} denotes the i-th internal node in the l-th level in S. E is a set of edges indicating the parent-child relationship between classes. Thus, there are three types of nodes in T: 1) *leaf classes* are known labels seen during training, 2) *internal classes* are ancestors of the leaf classes, which are also seen during training, 3) *novel classes* are unseen during training and only appear in T during inference. Note that leaf and novel classes are nodes without a child. Figure 1 shows an example in the Amazon dataset, where four representative product reviews of the classes "Educational Book", "CD Player", "Target Card Game", and "Chess Set" are listed at the leaf classes, respectively, while "Electronics for kids" and "Games" are internal classes and any other classes unseen during training, e.g., the reviews of products “DVD games” and “Camera camcorders” are classified as novel classes.

For a internal class s, let $C(s)$ denotes the set of known classes whose parent is s, $A(s)$ denotes the set of sś ancestors, and $N(s)$ denotes the set of novel classes whose closest known class is s. Note that $A(s)$ include s.

Given a text x and a taxonomy T, our goal is to predict the fine-grained class label y in T, which is either a leaf or a novel class. If a text is predicted as a novel class, we attempt to assign one of the internal classes, indicating that the text belongs to a novel class whose closest known class in T is that internal class.

4 Approach

We develop a novel model for hierarchical novelty detection, named DNA-HINT (Domain-Novelty Aware Hierarchical Introspection model). As shown in Figure 2, DNA-HINT consists of three components: (1) a semantic encoder to generate the representation of the input, (2) a domain-novelty aware network to calculate the domain-novelty aware score of each classifier as their confidence score in a top-down manner, (3) a hierarchical introspection network to compute a cross-entropy loss concerning the prediction errors level-wise.

4.1 Semantic Encoder (SE)

Following the finding from Hendrycks et al. (2020) that larger models are not always better for novelty detection tasks in NLP, we employ a pre-trained BERT Base model[1] as the encoder to generate the semantic representation of the input. Each input is tokenized and encoded with the BERT Base model. We use the output of the special $[cls]$ token as the semantic representation of the whole input sequence:

$$h = BERT(x) \tag{1}$$

where $h \in \mathbb{R}^k$ is the semantic representation encoded by BERT and k is the dimension of the word embedding..

4.2 Domain-Novelty Aware Network (DNAN)

Each internal class s_{li} has a threshold-based domain-novelty aware network that calculates the domain-novelty aware score f_{li} as its confidence score.

$$f_{li} = p(y|d_{in}, x) = \frac{p(y, d_{in}|x)}{p(d_{in}|x)} \tag{2}$$

where f_{li} denotes the derived DNA from internal node s_{li} and d_{in} is a binary domain variable. Specifically, f_{li} is derived by calculating the quotient of the domain-aware variable and the novelty-aware score as follows:

$$p(d_{in}|x) = \sigma(w_g h + b_g) \tag{3}$$

where σ is a sigmoid function, w_g and b_g represent the learnable parameters.

$$p(y, d_{in}|x) = w_h h + b_h \tag{4}$$

where w_h and b_h represent the learnable parameters. The decision rule for each s_{li} is defined as:

$$\tilde{y_s} = \begin{cases} \arg\max_{y'_s} P(y', d'_{in}|x, s) & \text{if confident,} \\ N(s) & \text{otherwise,} \end{cases}$$

where $P(y'_s, d_{in}|x, s)$ denotes DNA(s), $y'_s \in C(s)$, and $\tilde{y_s}$ is the predicted class. The top-down decision stops at s_{li} if the predicted class is a known leaf class or the classifier encounters an unconfident score. The confidence threshold that determines whether the classifier is confident enough is a class-dependent hyperparameter. Given the semantic representation, the internal classes are traversed according to the taxonomy in a top-down manner.

4.3 Hierarchical Introspection Network (HINT)

To generate a hierarchical representation, HINT first makes a two-step concatenation of the domain-novelty aware scores f produced by internal classes. Then, the hierarchical representation is used to compute a cross-entropy loss that introspects the prediction errors hierarchically. Specifically, the first concatenation is made level-wise to collect all domain-novelty aware scores f in the l-th level.

$$c_l = \oplus_{i=1}^{n_l}\{f_{li}\} \tag{5}$$

where c_l denotes the concatenation of domain-novelty aware scores in the l-th level, $\oplus$ denotes a concatenation operation, and n_l denotes the number of classes in the l-th level. Then, the second concatenation considers the levels above l and the l-th level to generate the hierarchical representation r_l.

$$r_l = \begin{cases} r_{l-1} \oplus c_l & \text{if } l \neq 1, \\ c_l & \text{if } l = 1, \end{cases} \tag{6}$$

where $l = 1$ denotes the root layer. The hierarchical representation r_l is then normalized by a

[1] https://pypi.org/project/pytorch-transformers/

	Amazon	DBPedia	20 Newsgroups	Reuter 52
# of level	4	4	2	2
# of leaf classes	505	173	15	44
# of internal classes	71	30	1	1
# data of known	47K	323K	14K	8K
# of novel classes	56	30	5	8
# data of novel	6K	19K	1K	889
# data per Train	30K	258K	7K	5K
# data per Dev	7K	32K	846	591
# data per Test	9K	32K	5K	2K

Table 1: Statistics of the datasets.

softmax function to generate the hierarchical prediction probability:

$$\tilde{y_{li}} = softmax_i(r_l) \quad (7)$$

where y_{li} denotes the hierarchical prediction probability for s_{li}.

The total loss aggregates the cross-entropy loss over layers according to the hierarchical prediction probability and ground truth class. We first define the loss of the l-th layer:

$$loss_l = -\sum_{j=1}^{n_l} y_{lj} log(\tilde{y_{lj}}) \quad (8)$$

where y_{lj} denotes the expected prediction of the j-th class in the l-th level. Finally, the total loss is derived by the summation of the loss over layers:

$$J(\theta) = \sum_{l=1}^{L} loss_l \quad (9)$$

where θ are the learnable parameters. We use Adam (Kingma and Ba, 2014) as the optimizer.

Among the three search methods proposed by Wu et al. (2016), we adopt the beam search method at training time to derive the hierarchical representation, while we implement the greedy method at test time.

5 Evaluation Setting

We evaluate the performance on four benchmark datasets. All datasets are in English language. For each dataset, we compile a training set and a test set that has additional novel classes. The training set is split into a sub-training set and a development set for validation.

For Amazon and DBPedia datasets, we expect a parent in the taxonomy to have at least one child as a novel class and two children as known classes, so we merge any class less than three children to obtain our tree-structured taxonomy. For example, if "Games" has only two children, one is "Card Games" and the other is "Board Games", we merge these three nodes as the "Games" node. For 20 Newsgroups and Reuters 52 datasets, we obtain a tree-structured taxonomy by adding the root on top of the existing classes. The dataset statistics are shown in Table 1.

For evaluation, we propose a new metric that gives appropriate credit for classification and novelty identification at each level. Evaluation results on Amazon and DBPedia are reported in terms of accuracy and our proposed metric. For the 20 Newsgroups and Reuters 52, they have a fairly flat taxonomy and are therefore reported using AU-ROC.

5.1 Datasets

Amazon[2][3] (He and McAuley, 2016) This dataset has six main products categories, such as "Toys Games", "Grocery Gourmet Food", and "Baby Products". We take 56 classes from each level as novel classes used during inference. Each review contains a textual review and a category (a leaf or novel class). This dataset is released without user's personal information.

DBPedia (Lehmann et al., 2015) This dataset consists of eight main Wikipedia article categories, such as "Agent", "Topical Concept", and "Sports Season". We take 30 classes from each level as novel classes used during inference. Each text is a summary of a Wikipedia article.

[2] `https://jmcauley.ucsd.edu/data/amazon/`

[3] `https://www.kaggle.com/kashnitsky/hierarchical-text-classification`

(a)
Zoo Med Laboratories SZMEE10 Eco Earth Pet Bed, 1-Brick.Eco Earth is perfect for my numerous frogs, green anoles and chameleon. I use It for the plants in side their habi's and for my house plants.

(b)
Xyla Apricot Jam, 10 Oz.Tastes absolutely fantastic. For low carb too many carbs but great for low fat diet. Could eat the whole jar but wont.

(c)
Cars.Yes, it seems false, because the other cars I bought from the same collection are different, seem stronger and the eyes of the car moves, as in 3D.

Ground Truth :	**Pet Supplies⇒Novel**	**Grocery Gourmet Food⇒Novel**	**Toys Games⇒Hobbies⇒Novel**
DNA-HINT :	**Pet Supplies⇒Novel**	**Grocery Gourmet Food⇒Novel**	**Toys Games⇒Novel**
MSP :	**Pet Supplies⇒Cats⇒Beds Furniture**	**Root⇒Novel**	**Root⇒Novel**

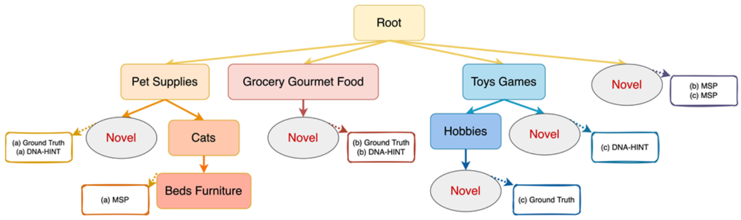

Figure 3: Qualitative results of hierarchical novelty detection in the Amazon dataset. Three test instances are demonstrated with the ground truth label and the predicted label of our DNA-HINT model and the MSP baseline model. Below demonstrates the partial taxonomy, where dashed edges denote the ground truth label and the prediction of the corresponding models and instances.

20 Newsgroups (Lewis et al., 2004) This dataset consists of 20 different newsgroup topics, such as "Autos", "Politics in the Middle east" and "Baseball". We randomly leave out 5 topic as novel classes in the test set.

Reuters 52 (Lang, 1998) This dataset has 52 main topics, such as "Jobs", "Livestock", and "Money Supply". 8 topics are randomly chosen as novel classes in the test set.

5.2 Evaluation Metric

For proper evaluation of hierarchical novelty detection, we propose a new metric to improve the inadequacy of existing metrics concerning the correctness for both the classification of each level and the identification of novelty.

For example in Figure1, misclassification into node "Electronic toys" (Toys Games⇒Electronics for Kids⇒Electronic Toys) when the true class is "Music Players Karaoke" (Toys Games⇒Electronics for Kids⇒Music Players Karaoke) should be punished less than misclassification into node "Board Games" (Toys Games⇒Games⇒Board Games) since the former case is in the same subtree while the latter is not [4]. Second, the hierarchical metrics are only able to judge hierarchical classification but not novelty identification. Third, optimizing the combination of confidence thresholds among the massive threshold-based classifiers in the taxonomy is not the goal of this paper to explore. Therefore, AUROC is not an expected metric for hierarchical novelty detection, especially for datasets with deep taxonomy.

To satisfy the requirements of hierarchical novelty detection, we proposed a new metric hnF_1 that combines the accuracy of novelty (Acc_{Novel}) and the hierarchical classification metric hF_1. Acc_{Novel} is calculated with standard accuracy, which each instance is only awarded when the predicted label and the gold label are both known classes or both novel classes. For example, if the true label is "Novel" (Toy Games⇒Novel) and the predicted label is "Novel" (Grocery Gourmet Food⇒Breads Bakery⇒Novel), then it's awarded for the score. hF_1 considers the class subset of ancestors for the ground truth label $A(y)$ and predicted label $A(\tilde{y})$ to calculate in a hierarchical manner. Then, the hnF_1 is computed as follows:

$$hnF_1 = \beta \cdot Acc_{Novel} + (1 - \beta) \cdot hF_1 \qquad (10)$$

where $\beta \in [0, 1]$ is a self-defined factor deciding the importance of novelty detection in the combined score. In this paper, all hnF_1 are reported with a β of 0.5. For example, assume the true label is "Eletronic Toys" (Toys Games⇒Eletronics for Kids⇒Eletronic Toys), we compare the performance for two misclassification cases, (a) "Music Players Karaoke" (Toys Games⇒Eletronics for Kids⇒Music Players Karaoke) gets 100% for Acc_{Novel} and

[4] ⇒ denotes the parent-child relationship in the taxonomy.

Model	Amazon			DBPedia			20 Newsgroups	Reuter 52
	Known	Novel	hnF_1	Known	Novel	hnF_1	AUROC	
MSP	70.97	18.86	68.89	74.06	2.09	65.46	77.51	93.36
DNA-HINT	71.24	19.63	69.69	76.43	2.97	66.59	78.67	93.79

Table 2: Hierarchical novelty detection results in the Amazon and DBPedia datasets. The novel accuracy is reported by searching the optimized thresholds.

66.66% for hF_1, so hnF_1 can be obtained as $0.5 \cdot 100\% + 0.5 \cdot 66.66\% = 83.33\%$; (b) "Novel" (Toys Games⇒Eletronics for Kids⇒Novel) gets 0.0% for Acc_{Novel} and 66.66% for hF_1, so hnF_1 can be obtained as $0.5 \cdot 0.0\% + 0.5 \cdot 66.66\% = 33.33\%$. Both (a) and (b) would get the same scores with existing metrics , i.e., 0% for accuracy and 66.66% for hf_1.

Besides our proposed metric, we also measure the area under known-novel class accuracy curves (AUC) presented by (Lee et al., 2018a). We obtain the AUC by varying all class-dependent thresholds with a fixed value, which aim to provide a more informative insight into the threshold independent performance.

5.3 Baseline

Hendrycks and Gimpel (2017) presented the maximum softmax probability (MSP) model as a baseline for novelty detection in various domains. Therefore, we choose the MSP model as our baseline for the hierarchical novelty detection task.

5.4 Implementation Details

The hyperparameter setting of all models is: word embedding dim=768, number of training epochs=100 with early stopping by 10 epochs, batch size=12, accumulate step=1, learning rate of the semantic encoder=1e-5, learning rate of each classifier=3e-4, optimizer=Adam. All models are executed on an Nvidia GeForce RTX 3090 GPU. As for the confidence threshold, which is a class-dependent hyperparameter, we adopt two search strategies in Appendix A.

6 Experiments

6.1 Results

Figure 3 show the qualitative results with test instances from the Amazon dataset. We observe that DNA-HINT can provide fine-grained predictions by utilizing the taxonomy of the dataset as expected. In Figure 3 (a), DNA-HINT correctly

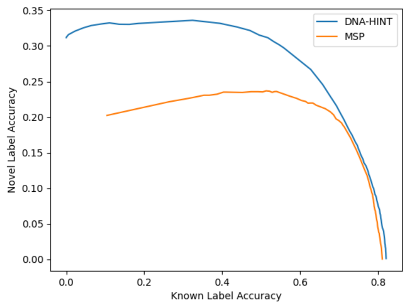

Figure 4: Known-novel class accuracy curves obtained by varying all class-dependent thresholds with a fixed value in the Amazon dataset for DNA-HINT and the baseline model.

finds the fine-grained label in the taxonomy, while the baseline classifies it as "Beds Furniture" (Pet Supplies⇒Cats⇒Beds Furniture), which not only incorrectly detects as a known class but also confuse the description of cats with reptiles. In Figure 3 (b), both of the models predict a novel class. As the ground truth class is "Jellies & Sweet Spreads" (Grocery Gourmet Food⇒Jams⇒Jellies & Sweet Spreads), which is a novel class of "Grocery Gourmet Food", DNA-HINT predicts a more informative label that finds the closest label in the hierarchy and the baseline only predicts it as a novel class of "Root". In Figure 3 (c), none of the models find the correct label, a novel class of "Hobbies". Compared to the baseline, DNA-HINT makes a closer prediction.

Table 2 shows that DNA-HINT outperforms the baseline on both Amazon and DBPedia datasets. The accuracy of the known class, the accuracy of the novel class and hnF_1 increased by 0.27%, 0.77% and 0.8% respectively on Amazon and 2.43%, 0.88%, and 1.13% respectively on DBPedia. Figure 4 exhibits the known-novel class accuracy curves on Amazon. The AUC is 23.77% and 14.60% for DNA-HINT and the baseline, respectively. DNA-HINT significantly outperforms the baseline.

The last two columns in Table 2 show the re-

Model	Accuracy			
	AUC	hnF_1	Novel	Novel at Level 4
our DNA-HINT	23.77	64.87	31.15	10.65
- DNAN	21.55	64.47	27.62	6.77
- HINT	19.44	62.64	27.20	7.90
- DNAN - HINT	14.60	60.89	23.68	6.41

Table 3: Ablation analysis on the test set of Amazon. The novel accuracy is reported with a guarantee of 50% known accuracy.

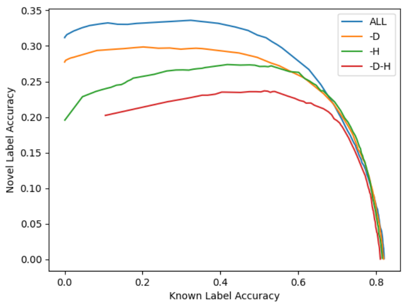

Figure 5: Known-novel class accuracy curves obtained by varying all class-dependent thresholds with a fixed value in the Amazon dataset for ablation analysis. "-D" denotes the removal of DNAN, "-H" denotes the removal of HINT, and "-D-H" denotes the removal of DNAN and HINT.

sults in 20 Newsgroups and Reuter 52, which have a fairly flat taxonomy and are therefore reported using AUROC. We observe that DNA-HINT outperforms the baseline significantly by 1.16% and 0.43% on 20 Newsgroups and Reuter 52, respectively. For both datasets, DNA-HINT also achieves substantial improvements by considering domain effects.

6.2 Ablation Analysis

To further illustrate the effectiveness of domain-novelty aware and hierarchical introspection networks, we conduct an ablation study on Amazon's test set. To observe the subtle changes that each component brings, Table 3 reports the performance where certain components were removed with a guarantee of 50% known accuracy. Among them, hnF_1 reflects the overall performance of the system in hierarchical novelty detection, and AUC reflects the comprehensive performance of the system's known-novel class accuracy under all parameters. Figure 5 further shows known-novel class accuracy curves for a more informative insight into the threshold independent performance with some components removed.

As expected, both AUC and hnF_1 continue to decrease with the removal of each component, demonstrating the effectiveness of binding DNAN and HINT. The last two columns in Table 3 show the accuracy of novel classes in total and at the lowest level that the actual category of the text inhabit. After removing DNAN, the accuracy drops by 3.88%, indicating that DNAN indeed improves the quality. After removing HINT, the lowest level drops significantly by 2.75%, demonstrating the importance of HINT's design for lower-level classification. From the results, we find that each component plays an important role, especially for the lowest-level detection that is most important in practical applications (e.g., e-commerce systems often use hierarchical classifications, where the lowest level represents the actual category of the text).

7 Conclusion

In this paper, we propose a new model for hierarchical novelty detection, the Domain Novelty-Aware Hierarchical Introspection model (DNA-HINT). DNA-HINT can distinguish text into finer-grained known and novel classes without problematic issues, including overconfidence, inconsistency, and blocking problems. We also design a new metric hnF_1 to accurately measure the combined performance of the model on both known and novel classes. On four benchmark datasets, DNA-HINT significantly outperforms the baseline and is particularly effective for the lowest-level detection that is most important in practical applications. In future work, we aim to add visual information to hierarchical novelty detection.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alex Freitas and Andre de Carvalho. 2007. A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Applications*.

Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep hierarchical classification for category prediction in e-commerce system.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517. ACM.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko and Fazel Famili. 2005. Functional annotation of genes using hierarchical text categorization. *Proceedings of BioLink SIG, ISMB*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*.

Ken Lang. 1998. Newsweeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339.

Kibok Lee, Kimin Lee, Kyle Min, Yuting Zhang, Jinwoo Shin, and Honglak Lee. 2018a. Hierarchical novelty detection for visual object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1034–1042.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018b. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Xiaoya Li, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. 2021. kFolden: k-fold ensemble for out-of-distribution detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3115, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. *arXiv preprint arXiv:1908.10419*.

Xipeng Qiu, Wenjun Gao, and Xuan-Jing Huang. 2009. Hierarchical multi-label text categorization with global margin maximization. In *Proceedings of the acl-ijcnlp 2009 conference short papers*, pages 165–168.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

A. Sun and E. Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, page 521, Los Alamitos, CA, USA. IEEE Computer Society.

Feihong Wu, Jun Zhang, and Vasant Honavar. 2005. Learning classifiers using hierarchically structured class taxonomies. In *Abstraction, Reformulation and Approximation*, pages 313–320, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing

Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

A Hyperparameter Search

The nature of hierarchical novelty detection is that there is no validation data of novel classes for hyperparameter search, which makes it difficult to choose the class-dependent confidence thresholds. We adopt two strategies, one is proposed by Lee et al. (2018a), which for each internal class s, a known leaf class that are not a descendant of s is recognized as a novel class.

$$\tilde{y_s} = \begin{cases} \arg\max_{y'_s} P(y', d'_{in}|x, s) & \text{if } P(\cdot|x, s) \geq \lambda_s, \\ N(s) & \text{otherwise,} \end{cases}$$

where λ_s is tuned with the harmonic mean of the accuracy between known and novel classes. Note that λ_s is a class-dependent hyperparameter for each internal class. We utilize this strategy to report the results on DBPedia.

The other strategy is sampling λ_s as a fixed value for all internal classes in the range of $[0.01, 1]$. We utilize this strategy to report the results on Amazon.