UDC 004.8
DOI: 10.15587/1729-4061.2025.321273

CONSTRUCTION OF HIERARCHICAL CLASSIFICATION MODEL FOR PRODUCT MANAGEMENT: PENALIZED INFORMATION GAIN CONSIDERING DYNAMIC WEIGHT COEFFICIENTS

Olga Narushynska
Corresponding author
Assistant*
E-mail: olha.o.narushynska@lpnu.ua
Maksym Arzubov
PhD Student, Assistant*
Vasyl Teslyuk
Doctor of Technical Sciences, Professor*
*Department of Automated Control Systems
Lviv Polytechnic National University
Stepana Bandery str., 12, Lviv, Ukraine, 79013

The object of this study is the process of constructing hierarchical classifiers for textual data within a defined taxonomy. The task addressed focuses on minimizing cascading errors and enhancing classification consistency across all hierarchy levels, a critical challenge for deep and imbalanced hierarchical structures. The proposed model leverages the Penalized Information Gain (PIG) criterion with dynamically adjusted weight coefficients.

A model for hierarchical text classification has been built. It aims to improve classification accuracy and preserve the structural logic of data within multi-level taxonomies.

Data featuring a multi-level taxonomy that meets classification requirements have been synthesized and are employed for training and testing classifiers. The performance of local and global hierarchical classification models was compared against conventional classifiers that do not account for taxonomic relationships between classes. The results demonstrate that using weight coefficients based on hierarchical levels enables an adequate representation of taxonomic dependences, which is crucial for maintaining data integrity and improving categorization quality at various levels. Experimental findings show an 8 % increase in the F1 score at the class and subclass levels.

A distinctive feature of the model built is the integration of dynamic weights into the splitting criterion, which allows hierarchical dependences between classes to be effectively addressed and cascading errors, typical of conventional approaches, to be minimized.

The model's practical application spans product management systems in e-commerce, text analytics in the restaurant business, and automated categorization systems for multi-level structures

Keywords: hierarchical classification, global and local model, cascading errors, weight coefficients, taxonomy

Received 04.12.2024
Received in revised form 20.01.2025
Accepted 10.02.2025
Published 21.02.2025

How to Cite: *Narushynska, O., Arzubov, M., Teslyuk, V. (2025). Construction of hierarchical classification model for product management: Penalized Information Gain considering dynamic weight coefficients. Eastern-European Journal of Enterprise Technologies, 1 (3 (133)), 17–27. https://doi.org/10.15587/1729-4061.2025.321273*

1. Introduction

The economy is becoming increasingly globalized, and trade is becoming one of the most important industries. The range of goods is very diverse, and in this regard, there is a need for hierarchical classification of goods [1]. It makes it possible to design multi-level structures that make it possible to quickly and efficiently organize goods into certain categories and subcategories.

Modern tasks of automatic data classification increasingly go beyond simple classifiers, requiring the construction of complex hierarchical structures that reflect the natural relationships between categories [2]. Hierarchical classification, unlike flat, not only makes it possible to take into account the relationships between classes but also opens up the possibility of creating classifiers that reflect the logic of the subject area. The importance of hierarchical models increases under conditions where the data has a tree-like structure, and errors at the upper levels can significantly affect the final result in subcategories [3].

Despite their promising potential, hierarchical methods often face challenges such as balancing the weights of different levels of the tree, minimizing cascading errors, and optimizing the separation criterion [4]. One of the key tools to address these issues is Penalized Information Gain (PIG), which combines statistical informativeness with taxonomic structure. However, the unique challenge is to build a model that gives more weight to higher levels of the hierarchy and gradually reduces their importance at lower levels, which ensures both accuracy and logical consistency of classification [5].

Of particular interest is the comparative analysis of the performance of hierarchical classifiers with weight coefficients and conventional flat (general) classifiers (flat classifier). While flat models are technically simpler, they ignore structural relationships, which can lead to inconsistency of the results of the subject domain. The hierarchical model, on the other hand, is able to preserve the semantic integrity and structural consistency of the classification results.

The relevance of the topic is predetermined by the need to build a new generation of categorization models that not only effectively solve the problem of accurate prediction but also naturally reflect the structure of real hierarchical data.

2. Literature review and problem statement

In practical applications such as commodity categorization systems, text analytics or managing complex taxonomies, such models are key to ensuring accuracy, clarity, and logicality of results [6]. To address this problem, machine learning models for hierarchical commodity classification have been investigated, taking into account the global standard GS1 Global Data Model [7]. This standard defines general principles and rules for commodity classification, which are used in more than 150 countries around the world.

Hierarchical classification involves the distribution of objects in a tree or directed acyclic graph (DAG), where each class belongs to a specific level. An important challenge is balancing errors at the upper and lower levels. In [1], a comprehensive overview of models is provided, which are divided into local and global models.

Local models build separate classifiers for each node, level, or branch of the hierarchy. However, they often suffer from cascading errors, when errors at the upper levels propagate to the lower ones. Global models involve building a single classifier for the entire tree and immediately predicting the class according to the hierarchy, as demonstrated in study [8]. Their disadvantage is the reduction in accuracy due to the failure to take into account local dependences between classes. The issue of balancing the advantages of local and global approaches remains unresolved, which is a promising area of research.

One of the early studies in the field of hierarchical classification is reported in paper [9], in which the author proposes a method for building a model using the support vector method and using multidimensional sparse text representation.

Work [10] provides an overview of scientific research in the field of hierarchical classification in various application areas, such as natural language processing, computer vision, bioinformatics, etc. The authors propose generalized metrics that take into account the "severity" of errors at different levels of the tree. The evaluation of such models requires a balance between accuracy, completeness (recall), and preservation of semantic consistency of the classification.

In [11], the problem of categorizing objects in complex hierarchical structures, when some of the labels in the training data are unobserved, is considered. The proposed probabilistic model makes it possible to compensate for the lack of labeled examples by learning from several sources. However, this approach may have limitations when working with deep hierarchies where the accumulation of cascading errors significantly affects the final classification accuracy.

Study [12] uses hierarchical multiclass classification for automatic text categorization, proposing a method for assessing the quality of classification based on the confusion matrix and accuracy metrics. However, the issue of preserving semantic consistency between hierarchy levels remains open due to the limitations of the metrics used.

The key challenges in hierarchical classification are cascading errors, the problem of balancing the weights of different hierarchy levels, uneven distribution of data in different branches of the taxonomy, and the lack of a universal approach to optimizing the separation criteria. The problem considered in [13] is the reduction of classification accuracy due to the cascading accumulation of errors in the process of passing through hierarchical levels. In the paper, the authors consider the issue of hierarchical classification with multiple labels that can be linked in a hierarchical structure. The authors propose the use of decision trees for multi-level classification, but the method does not completely solve the problem of balancing weights at different levels of the hierarchy and maintaining semantic consistency between classes.

The development of models based on deep neural networks has significantly improved the accuracy of hierarchical classification. The study in [14] focuses on solving the problem of adapting BERT-based models to hierarchical categorization tasks. The modified BERT for multi-level product classification uses product names and descriptions to train contextual representations, providing high accuracy in tasks with a hierarchical structure. However, despite the use of contextual representations of product names and descriptions, the model may show reduced performance in cases where the classes are deeply nested or have imbalances in representation.

The study reported in [15] addresses the problem of maintaining consistency of predictions in deep hierarchical structures. Despite the proposed model integrating hierarchical tree information and implementing a loss function to penalize inconsistencies, optimizing this function for different types of hierarchies remains a challenge.

In addition, study [16] focuses on the difficulty of matching different neural architectures to effectively account for hierarchical dependences. The effectiveness of combining different neural architectures (deep autoencoders and DBN) for tasks with multi-level distribution of goods in a large e-commerce domain has been demonstrated.

There are also studies that use deep neural networks to solve the problem of hierarchical classification of short texts. For example, in [17], the problem of effective use of the attention mechanism to take into account deep hierarchical dependences is considered, since with increasing levels of hierarchy the model may lose the consistency of predictions and experience difficulties with training on unbalanced data.

Paper [18] emphasizes the difficulties of maintaining semantic consistency across hierarchy levels when using distributional semantics, as vectorization methods such as Word2Vec may not accurately capture taxonomic dependences when contextual differences between classes are minimal or when the data are unevenly distributed.

Key challenges and shortcomings of existing models:

1. Cascading errors: errors at the upper levels of the tree propagate to the lower levels, which is a significant problem for local methods.

2. Uneven data distribution: data is often concentrated in certain branches of the tree, which leads to difficulties in training models [19].

3. Ignoring structure: flat classifiers, although fast and simple, do not take into account the hierarchy, which makes them less effective for problems with complex taxonomies [15].

4. Weight balance: in modern models, the influence of weights for different levels of the hierarchy has not been sufficiently studied, which is a promising area of research.

It has been found that flat classifiers do not take into account the hierarchical structure, which leads to cascading errors and loss of semantic consistency. Hierarchical approaches, on the contrary, provide better consistency, but often face problems of data unevenness and high computational costs. Thus, our review of the literature [8–15] revealed relevant research directions: the construction of a model capable of combining the advantages of local and global approaches with the minimization of cascading errors, optimization of weight coefficients, and preservation of semantic consistency between classes.

3. The aim and objectives of the study

The aim of our research is to build a hierarchical categorization model for text data that improves classification accuracy and preserves the hierarchical structure of data by using variable weights in the separation criteria. This will minimize cascading errors, ensure consistency of classification results, and improve the efficiency of models in tasks with multi-level structures.

To achieve the goal, the following tasks were set:

– to synthesize a training dataset that takes into account the structure of a multi-level taxonomy of text data;

– to develop a scheme for a hierarchical classification algorithm based on PIG to preserve the hierarchical logic of data;

– to implement an experimental model for testing on hierarchically structured text data;

– to evaluate the effectiveness of the implemented model in tasks with different levels of taxonomic complexity and compare it with conventional hierarchical categorization models.

4. The study materials and methods

The main hypothesis of the study assumes that the PIG criterion with dynamically changing weights makes it possible to increase the accuracy of text data classification and to ensure consistency between hierarchy levels, which reduces cascading errors typical of existing models.

Assumptions adopted in the study:

1. Each text description belongs exclusively to one class at each hierarchy level, which is defined by business requirements.

2. The hierarchical data structure accurately reflects the real taxonomic dependences used in classification tasks.

3. Semantic relationships between classes represented in text data are adequately reflected as a result of preprocessing and vectorization.

4. Classes in each node of the hierarchy are mutually exclusive, that is, each object can belong to only one subcategory within a given level.

Simplifications adopted in the study:

1. The hierarchical data structure is considered static, without changes in the taxonomy during training and testing.

2. The number of hierarchy levels is known in advance, and all objects have clear labels for each level.

A hierarchical taxonomy is an organizational structure used to describe and categorize objects, such as things, ideas, or living organisms. In such a structure, objects are placed into groups that reflect their degree of similarity and difference [21]. Each group may contain subgroups that describe the relationships between objects in more detail.

In some cases, when the categories are complex and broad, a hierarchical classification can be more effective than a single-category categorization.

A hierarchical product taxonomy is the organization of products into a logical hierarchy of classes, with more general categories at the top and more specific categories at the bottom. This hierarchical class structure helps organize a large number of products and simplifies the process of searching and categorizing products for users. A taxonomy can be built based on various criteria, such as functionality, material, manufacturer, or type of consumer [22]. For example, in a household taxonomy, more general categories might include "Electronics", "Furniture", "Crockery", and more specific categories might include "Refrigerators", "Bedrooms", "Kettles".

A single product may be included in multiple classes of a hierarchical taxonomy. For example, if a product taxonomy includes the categories "Fruits" and "Exotic Fruits", then the fruit "Mango" may be classified as a product in the category "Mango" of the subcategory "Exotic Fruits", as well as a product in the category "Fruits". However, each specific hierarchical taxonomy may have different rules about whether products can be categorized in multiple categories.

The study considers the case where a product can be assigned to only one class at each level of the hierarchy. For example, if the first level has the classes "Clothing", "Electronics", and "Books", then the product can belong to only one of these classes, and its further classification will be carried out within the selected class.

Hierarchical classification is the process of determining the class to which an object belongs, taking into account its belonging to a higher level of abstraction. In hierarchical classification, there are several levels of classes, where higher levels are general classes, and lower levels are more specific classes. Thus, hierarchical classification makes it possible to create a "parent-child" relationship between classes, where higher classes (parents) have more general characteristics, and lower classes (sons) have more specific characteristics. This makes it possible to use the class structure to increase classification accuracy and reduce the number of errors.

The study is based on text data containing a multi-level hierarchical class structure. Each object belongs to a specific node in the hierarchy, and the hierarchical tree includes several levels of categories that represent general and specific classes.

Two types of models were used to categorize data:

– local models: separate classifiers are built for each level or node of the hierarchy. In particular, the Level-Wise Local method is used, where a separate classifier is built for each level of the hierarchical tree, which predicts the belonging of the object to the categories at the corresponding level;

– global models: a single classifier is built that simultaneously takes into account all levels of the hierarchy. In this context, the global model is considered a flat classifier since the data structure is processed as a single task without building separate models for each level.

A global model was built using the Penalized Information Gain (PIG) criterion and dynamically variable weight coefficients. For this purpose, weight variables were introduced that take into account the level of the node in the hierarchical tree. The weight function is defined as follows:

$$W_i = \frac{W_0}{level_i}, \qquad (1)$$

where W_0 is the base weight for the hierarchy level and $level_i$ is the depth of node i in the hierarchical tree.

This provides greater significance to the upper levels of classification, minimizing the impact of errors at lower levels, which is key to preserving the structural logic of the data.

Penalized Information Gain (*PIG*) is an improved metric for evaluating partitions in tree models, specifically designed to take hierarchical relationships into account in the decision-making process. While conventional Information Gain (*IG*) focuses solely on reducing the entropy (uncertainty) in the sample, *PIG* extends this metric by adding a penalty term based on Taxonomic Informativeness (*TI*). This improvement ensures that splitting takes into account the hierarchical structure of the data, favoring those that better separate taxonomically distinct classes.

PIG integrates *TI* into the *IG* framework through a Penalty Factor (*PF*), which scales *IG* depending on the taxonomic relevance of splitting:

$$PIG(S)=IG(S)\cdot PF, \tag{2}$$

where the Penalty Factor (*PF)* is defined as:

$$PF=\log(1+\alpha\cdot ATI), \tag{3}$$

where α is a scaling factor that controls the sensitivity of PF to changes in Average Taxonomic Informativeness (ATI); the logarithm provides a nonlinear relationship that ensures that PF increases gradually with increasing ATI.

The logarithmic function ensures that PF changes gradually, avoiding too sharp transitions. PF increases with increasing ATI, rewarding the classification that better fits the hierarchy. This formula ensures that the Penalized Information Gain (PIG) reflects both the statistical significance of splitting and the hierarchical consistency of predictions.

5. Results of research on the hierarchical categorization model

5. 1. Synthesis of a training dataset that takes into account the structure of a multilevel taxonomy of text data

To conduct research into hierarchical classification, taxonomy from [7] was used. This hierarchy was chosen because of its prevalence in various industries, as well as because it has a sufficiently large number of classes and subclasses, which allowed for a thorough analysis of different hierarchical categorization models.

The class structure in GS1 GPC [7] was developed to categorize goods and services to simplify their exchange between partners in the supply chain. This hierarchical structure consists of 4 levels: segment, family, class, and subclass. At the first level, there are 21 segments, each of which is divided into several families. At the second level, there are more than 200 families, each of which includes different classes at the third level. Finally, at the fourth level, there are more than 5,000 subclasses, which are divided into specific products and services. The classification is based on grouping goods and services according to their characteristics, such as manufacturer, functional purpose, physical properties, and other criteria, which makes the classification process more accurate and simpler.

Only some product segments were selected for the study: Food/Beverages/Tobacco, Cleaning/Hygiene, Tableware, and Cutlery. These categories were selected to focus on popular and widely used products. Part of the class hierarchy tree is shown in Fig. 1.

The work is based on the use of a dataset containing a description of products and their belonging to a hierarchical classification.

The first dataset used in the work contains data on products of a company that is a supplier to the restaurant business. These data were manually labeled by experts according to a taxonomy that was created based on the hierarchical classification [7]. The second dataset used is data from the directionsforme website. These data contain a description of products and their belonging to the same hierarchical classification used in the first dataset. The use of such data makes it possible to compare results obtained from different sources and increase the reliability of the work.

To ensure the integrity of analysis, both datasets were combined into a single data set (Table 1). This was possible due to the identical structure of the class hierarchy and the similarity of product descriptions in both sources. This approach makes it possible to compile a more representative data set for research in the field of hierarchical product classification and provides expanded opportunities for the application of machine learning methods [22].

Therefore, the integration of these datasets not only exploits their complementary features but also contributes to the deepening of research in the field of hierarchical classification of goods, providing higher reliability and broader analytical potential.

The input data for the model are text descriptions of objects, an example of which is given in the column "Product Description" (Table 1). The data is preprocessed using Byte Pair Encoding (BPE) and transformed into vector representations based on BERT [24]. These vectors include semantic information that preserves key features of the text necessary for further analysis. Additionally, the model takes into account the hierarchical structure of the data, where each object has labels corresponding to different levels of taxonomy, in particular "Segment", "Family", "Class" and "Block".

The result of the model's work is the prediction of appropriate classes for each level of the hierarchy, which not only correspond to the input text but also comply with the structural logic of the hierarchy. In particular, the output includes labels that accurately reflect the object's belonging to categories at higher levels (e.g., "Segment") and at the same time detail information at lower levels (e.g., "Block"). This ensures both classification accuracy and maintaining semantic consistency between classes.

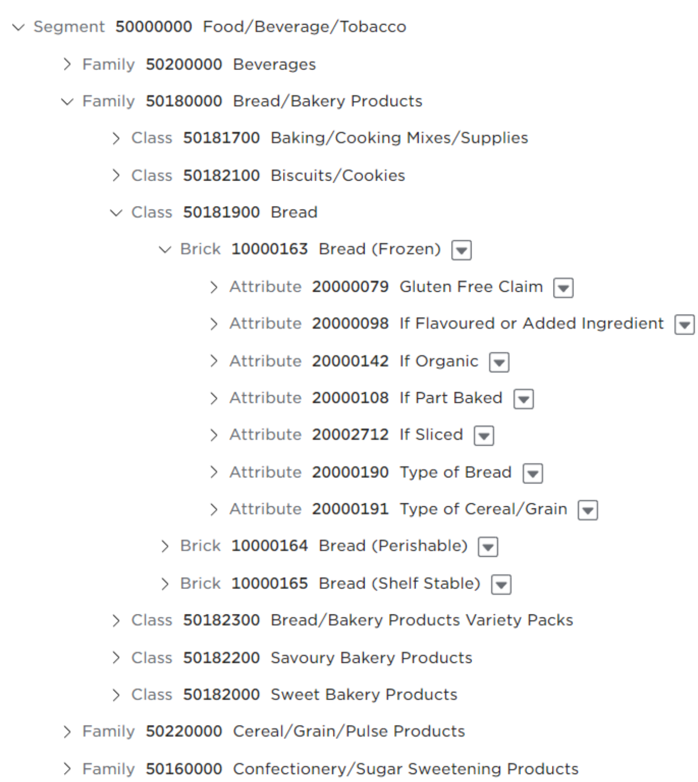

Fig. 1. Hierarchy of product classes [23]

Table 1

Some examples from the synthesized dataset for training and testing models

Product description	Segment	Family	Class	Brick
Village Candle Candle 1 ea JAR	Cleaning products	Cleaning products	Fresheners/ deodorisers	Deodorisers (non powered)
Yankee Candle Scenterpiece Easy Melt-cup 2.2 oz JAR	Cleaning products	Cleaning products	Fresheners/ deodorisers	Deodorisers (non powered)
Cascade Dish-washer Deter-gent 90 oz BOX	Cleaning products	Cleaning products	Dish care	Automatic dishwasher – detergent
Ahold Automatic Dishwashing Gel Lemon Scent	Cleaning products	Cleaning products	Dish care	Automatic dishwasher – detergent
Cravings by Chrissy Teigen Au Gratin Preseasoned Cast Iron 1 ea CARD	Kitchen-ware and tableware	Kitchen-ware	Cookware/ Bakeware	Bakeware/ ovenware/ grillware (non disposable)
Grillmark Scoop 1 ea NOT PACKED	Kitchen-ware and tableware	Kitchen-ware	Cookware/ Bakeware	Bakeware/ ovenware/ grillware (non disposable)
Duncan Hines Creamy Butter-cream Premium Frosting	Food/ beverage	Bread/ bakery products	Baking/ cooking mixes/sup-plies	Baking/cook-ing supplies (shelf stable)
Bobs Red Mill Flour 20 oz POUCH	Food/ beverage	Bread/ bakery products	Baking/ cooking mixes/sup-plies	Baking/cook-ing supplies (shelf stable)
Certo Premium Liquid Fruit Pectin, 2 ct – 6.0 oz Box	Food/ beverage	Bread/ bakery products	Baking/ cooking mixes/sup-plies	Baking/cook-ing supplies (shelf stable)
Pillsbury Cookies Chocolate Chips	Food/ beverage	Bread/ bakery products	Biscuits/ cookies	Biscuits/ cookies (per-ishable)
Lillys Baking Company Cookies 8 oz CLAM SHELL	Food/ beverage	Bread/ bakery products	Biscuits/ cookies	Biscuits/ cookies (per-ishable)
Voortman Cookies Holiday Treats	Food/ beverage	Bread/ bakery products	Biscuits/ cookies	Biscuits/ cookies (shelf stable)
Maro Polo Golden Rusk Toast	Food/ beverage	Bread/ bakery products	Bread	Bread (shelf stable)

5. 2. Construction of a scheme for a hierarchical classification algorithm based on PIG to preserve hierarchical data logic

A scheme for a decision tree construction algorithm that takes into account the hierarchical data structure is proposed (Fig. 2). The scheme is based on the use of the Penalized Information Gain (PIG) criterion with dynamically changing weights to optimize the separation process between classes at each level of the hierarchy.

The algorithm is aimed at ensuring a balance between statistical optimization of separation and preservation of logical consistency of classification at different levels of taxonomy. The key feature of the scheme is the adaptation of weight coefficients depending on the hierarchy level, which makes taking into account the importance of each level in the classification structure.

The proposed scheme can be used as the basis for a hierarchical categorization model to implement tasks related to minimizing cascading errors and improving the quality of classification in multi-level data structures.

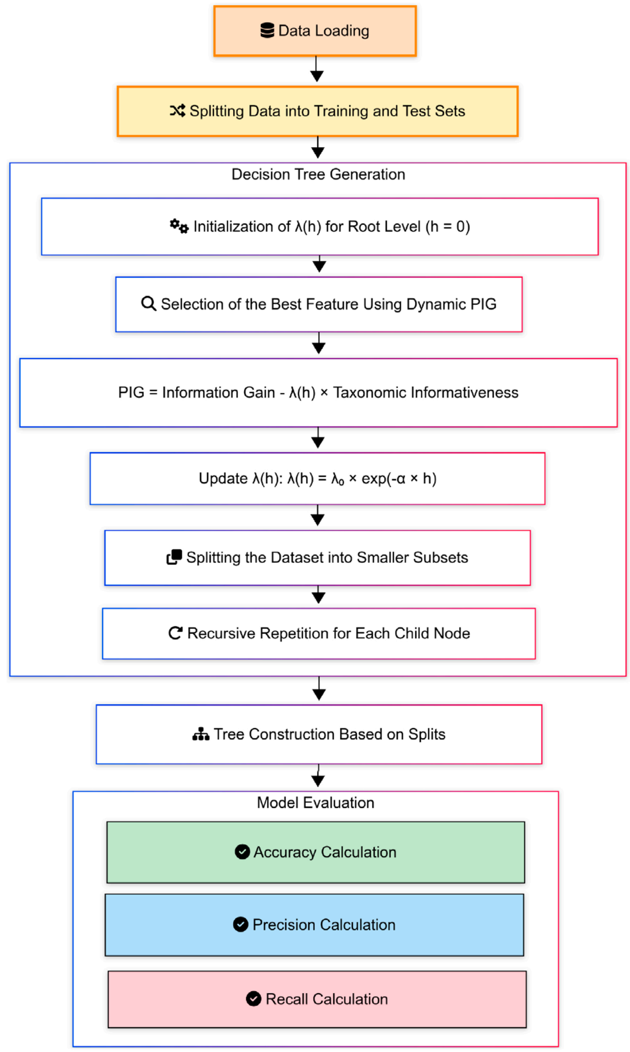

Fig. 2. Schematic of the decision tree construction algorithm with dynamic Penalized Information Gain (PIG), which takes into account the change in the weight coefficient depending on the level of the tree

The diagram depicts the process of constructing a decision tree using the dynamic Penalized Information Gain (PIG) criterion, where the weighting factor decreases as the depth of the tree increases, ensuring optimal separation at each level of the hierarchy.

5. 3. Implementation of an experimental hierarchical categorization model for testing on hierarchically structured text data

The Python programming language was used to implement the categorization model, which is one of the most common in the field of machine learning due to its flexibility and wide selection of libraries. The scikit-learn (sklearn) library, which is a standard for many machine learning tasks, became the basis for building a hierarchical categorization model. This library provides a powerful toolkit for working with models based on decision trees. However, its standard implementations have limitations in terms of modifying the criteria for dividing tree nodes. Since the developed model involves the use of its own division criterion – Penalized Information Gain (PIG), the implementation of the model requires modifying the source code of the sklearn library.

To do this, the process of creating a copy of an existing software project (for example, a library or code repository) was performed in order to make own changes or modifications. This term is often used in the world of software development, especially when working with open source. This allowed for the following:

– to integrate our own separation criterion, which takes into account dynamic weighting coefficients depending on the hierarchy level;

– to adapt the decision tree construction algorithm to hierarchical classification tasks and use this algorithm to build and train a random forest model;

– to maintain compatibility with other components of the scikit-learn ecosystem, such as data preprocessing and model validation methods.

The hierarchical categorization system is implemented in the form of an API (Application Programming Interface), which provides integration of the model into various software environments and automates the process of categorizing text data. The API is an important component that makes it possible to to evaluate the effectiveness of the devised decision tree construction algorithm scheme. The developed API supports the POST method, which accepts JSON data with information about products (Fig. 3).

Each data element contains:

– item_id – a unique product identifier;

– item_name – a text description of the product to be categorized.

The API returns a JSON response (Fig. 4), which includes:

– text description of the product (product_description);

– hierarchical classification categories (segment, family, class, brick);

– probability of belonging to the category (probability), which is an indicator of the model's confidence in the obtained result.

Curl

```
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '[
  {
    "item_id": 1,
    "item_name": "Yankee Candle Reed Diffuser 1 ea BOX"
  },
{
    "item_id": 2,
    "item_name": "Good Molly'\''s Brownie Bites 4.5 oz STAND PACK"
  }
]'
```

Request URL

```
http://localhost:8000/predict
```

Fig. 3. Example of an HTTP POST request to API for categorizing products by hierarchical structure, executed in the Postman system using a local categorization model server

Server response

Code Details

200

Response body

```
[
  {
    "id": 1,
    "product_description": "Yankee Candle Reed Diffuser 1 ea BOX",
    "segment": "Cleaning/Hygiene Products",
    "family": "Cleaning Products",
    "class": "Fresheners/Deodorisers",
    "brick": "Air Fresheners/Deodorisers (Non Powered)",
    "probability": 0.96
  },
  {
    "id": 2,
    "product_description": "Good Molly's Brownie Bites 4.5 oz STAND PACK",
    "segment": "Food/Beverage/Tobacco",
    "family": "Bread/Bakery Products",
    "class": "Baking/Cooking Mixes/Supplies",
    "brick": "Baking/Cooking Supplies (Shelf Stable)",
    "probability": 0.92
  }
]
```

Download

Response headers

```
content-length: 494
content-type: application/json
date: Wed,05 Feb 2025 15:52:53 GMT
server: uvicorn
```

Fig. 4. Server response to an HTTP POST request sent through the Postman system, which displays the results of hierarchical product classification

The results of the API work confirm that the devised algorithm scheme could be effectively implemented in hierarchical categorization systems. The accuracy, scalability, and flexibility of API make the proposed approach suitable for industrial use in tasks of automating the classification of complex hierarchical data structures.

5. 4. Comparative analysis of the built model with conventional hierarchical categorization models

The study used several approaches to classification, including flat classifiers (Flat Classifier), which do not take into account the hierarchical structure, and hierarchical approaches: step-by-step by levels (PerLevel), step-by-step by nodes (PerNode), and step-by-step by parent nodes (PerParent) [25]. To implement these approaches, popular machine learning algorithms were used – Random Forest and XGBoost. In addition, a global model was built with the implementation of the dynamic separation criterion Penalized Information Gain (PIG), which makes it possible to take into account the hierarchical structure of data at all levels. The global approach provides simultaneous optimization of separation for all levels of the hierarchy, which minimizes cascading errors and improves classification consistency. The models were compared using the F1-metric indicator for all levels of the hierarchy of the input dataset, which is shown in Fig. 5–8, and confirms the effectiveness of our approach.

The basic method for preparing text data is the integration of Byte Pair Encoding (BPE) [26] with BERT vector representations [27]. This method makes it possible to convert text descriptions into numerical vectors, preserving their semantic content, in particular in cases of abbreviations or ambiguities in the text.

Additionally, the input data was standardized using StandardScaler to ensure uniformity of the scales of numerical features, which helps improve the performance of machine learning models.

Flat models were based on the use of models that do not take into account the hierarchical structure of the data [8]. For this purpose, two main models were used: Random Forest [3] and XGBoost [28], tuned with optimal hyperparameters for the classification problem. In addition to the standard separation criterion, flat models were also integrated with the developed dynamic PIG criterion. This model makes it possible to take into account the relationships between classes, improving the learning process by penalizing incorrect splitting of tree nodes that do not correspond to the semantic structure of the taxonomy [29].

Hierarchical models involved the use of local classifiers at different levels of the hierarchy. Several strategies were implemented, including classification for each node (PerNode), each level of the hierarchy (PerLevel), and for each parent node (PerParent). These models used local versions of Random Forest and XGBoost, tuned to ensure maximum performance. Local classifiers had the advantage of taking into account local contexts and relationships between classes at different levels of the hierarchy, which is critical for the multilevel classification problem [29].

The model parameters were optimized to maximize classification accuracy. To assess the generalization ability of the models, the data were divided into a training (80 %) and a test (20 %) sample. The test sample contained only those categories that were represented in the training set to avoid the negative impact of unknown categories on the classification results (Table 2).

Table 2

Distribution of unique categories and data volumes in training and test samples of hierarchical classification in the study

Sample	Number of unique Segments	Number of unique Families	Number of unique Classes	Number of unique Blocks	Sample size
Training	3	17	44	89	28,677
Test	3	17	44	89	7,201

The evaluation of the results of model predictions was carried out for each level of the hierarchy (Segment, Family, Class, Block), which allowed us to identify the peculiarities of the models' work at different levels of detail. To display the results and analyze the models' work, learning curves were constructed using the Weights and Biases (wandb) platform, which allowed us to track the progress of training and identify possible signs of overtraining.

The experimental results (Fig. 5–8) demonstrate a significant difference in performance between flat (Flat) and hierarchical categorization models, as well as a noticeable effect of the dynamic criterion Penalized Information Gain (PIG) on increasing the accuracy of the models. The efficiency was assessed using the F1-metric at all levels of the GPC hierarchy: Segment, Family, Class, and Block. The visualizations in the figures clearly show how different models affect the classification accuracy at each level.

Flat models showed moderate results, especially at deep levels of the hierarchy (Class and Brick). In particular, Flat Random Forest shows limited generalization ability due to the fact that this model does not take into account the hierarchical nature of the data. This is confirmed by its relatively low F1-metric at the Brick level, which is the most detailed level of the taxonomy.

Flat XGBoost, while outperforming Random Forest at all levels, also has limitations related to ignoring the hierarchical structure. The lack of contextual relationships between classes leads to classification errors, which are especially noticeable at the Brick and Class levels.

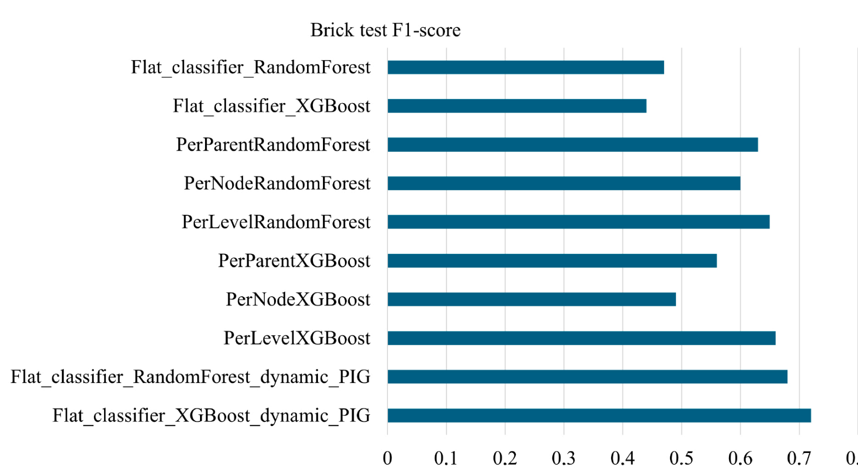

Fig. 5. Comparison of F1-metric of categorization models at the lower level of the Brick hierarchy

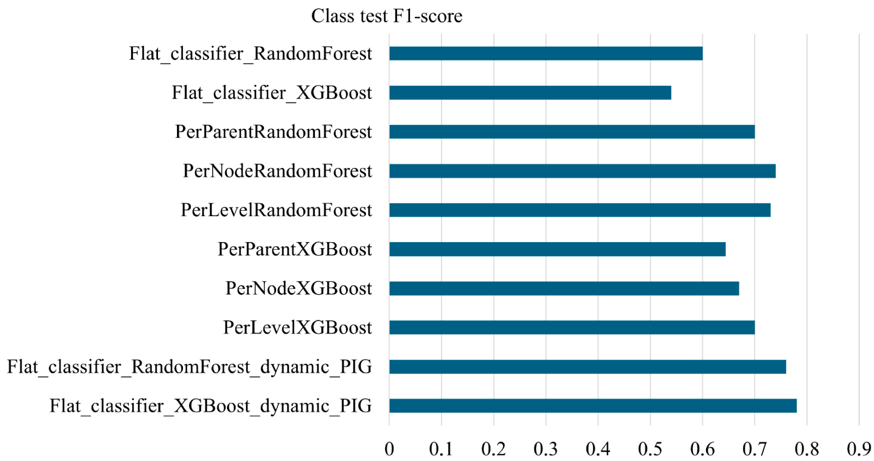

Fig. 6. Comparison of F1-metric of categorization models at the Class hierarchy level

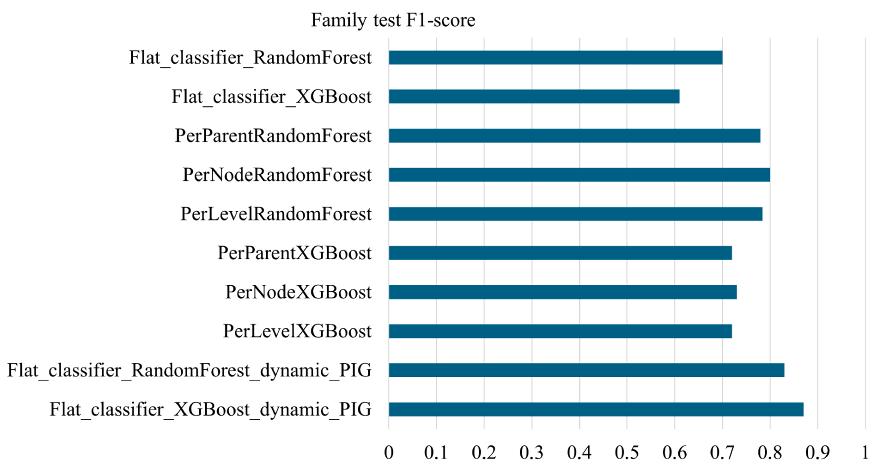

Fig. 7. Comparison of F1-metric of categorization models at the Family hierarchy level

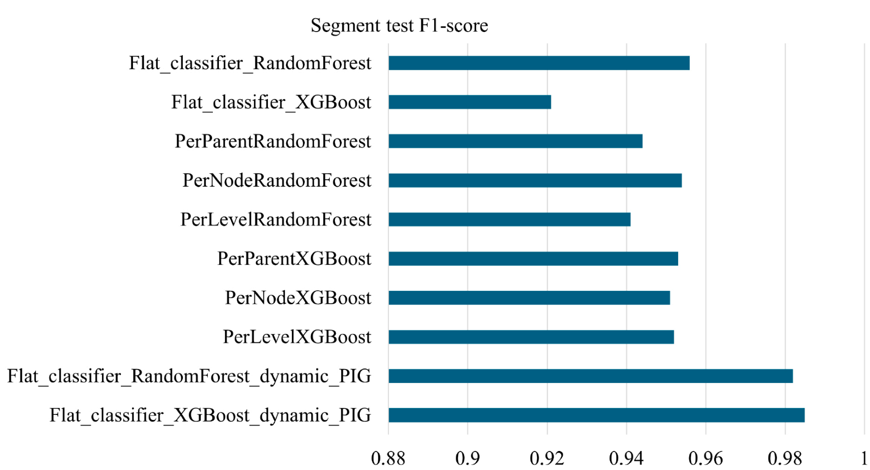

Fig. 8. Comparing the F1-metric of categorization models at the top level of the Segment hierarchy

It is important to note that the integration of dynamic PIG into these models significantly changes the situation. Flat Random Forest and Flat XGBoost with dynamic PIG show significant improvement, especially at the Brick level, where the F1-metric increases to 0.72 for XGBoost. This indicates that dynamic PIG effectively compensates for the weaknesses of flat models, ensuring that the classification is consistent with the hierarchical structure.

Hierarchical models showed significantly better results compared to standard flat models. The highest performance among hierarchical models is demonstrated by PerLevel, which takes into account the specificity of each level of the hierarchy. For example, for the Class and Brick levels, the F1-metric exceeds 0.65 for PerLevel Random Forest and XGBoost. This indicates that taking into account the local context of each level makes the model to identify classes more accurately.

The PerParent model showed slightly lower results since its performance depends on the quality of classification at higher levels of the hierarchy. This means that errors that occur at the Segment or Family levels propagate downward, affecting the classification accuracy at the Class and Brick levels.

The PerNode model, although showing competitive results, faces the problem of overfitting at individual nodes of the hierarchy. This may explain why its performance is less stable compared to PerLevel.

The integration of dynamic PIG into all types of models allowed us to significantly improve their performance.

For example, XGBoost with dynamic PIG achieved an F1-metric of 0.72 at the Brick level, the highest among all the models tested.

This supports the hypothesis that considering hierarchical dependences in the training process helps improve classification accuracy, especially for deep levels of the hierarchy.

An important observation is that even at the Segment level, where there are fewer classes, using dynamic PIG helps improve accuracy. This suggests that PIG not only improves classification at deep levels but also provides a more balanced performance of the models throughout the hierarchy.

6. Discussion of results based on investigating the hierarchical classification of text data

The synthesized dataset (Table 1) reproduces real conditions under which the model has to cope with the uneven distribution of data between categories and the deep structure of the hierarchy. The use of a training dataset with more than 35 thousand examples covering the multi-level structure of the GS1 GPC hierarchy (Table 2) ensured the accuracy and consistency of the classification approaches.

The built scheme of the decision tree construction algorithm using dynamic weight coefficients (Fig. 2) makes it possible to reduce cascading errors: dynamic weight reduction at lower levels of the hierarchy maintained the consistency of predictions between segments, families, and subclasses. The model with dynamic PIG at the Brick level (the lowest level of the hierarchy) ensured stable prediction, preventing the propagation of errors from the upper levels.

The designed API provides convenient integration of the hierarchical categorization model into software environments, simplifying the automation of the text data classification process. In particular, the API supports HTTP requests that make it possible to receive category predictions in JSON format (Fig. 3, 4), which contains a text description, hierarchical labels (segment, family, class, brick), and membership probabilities, providing flexibility for use.

Analysis of our results (Fig. 5–8) revealed that flat models (flat classifiers) have significant limitations in taking into account the hierarchical structure of the data, especially at deep levels. For example, flat XGBoost at the Brick level has a significant decrease in accuracy, reaching only 0.44 in terms of F1-metric. In contrast, the use of dynamic PIG in XGBoost increased the F1-metric to 0.72, which is 28 % higher.

Comparative analysis also revealed a significant impact of dynamic PIG on classification consistency. For example, at the Segment level, which contains only a few classes, even flat models with PIG showed a stable improvement in F1-metric to 0.98 compared to standard flat models, which reached a maximum of 0.95. This confirms that PIG effectively takes into account the hierarchical structure at all levels, even for large product groups.

The models built according to the PerNode principle, although they showed good results at the upper levels of the hierarchy, still lost significantly at the Brick and Class levels due to the accumulation of errors. For example, XGBoost with PIG at the Class level reached 0.78, while PerNode XGBoost only achieved 0.67.

The results demonstrate the effectiveness of the hierarchical categorization model built, which uses the dynamic criterion Penalized Information Gain (PIG). This result is explained by the fact that dynamic weights take into account the importance of the hierarchy levels, gradually reducing the weight value at deeper levels. This contributes to the effective distribution of significance between different levels of the hierarchy, which allows the model to take into account both local and global dependences.

The main advantage of our model is its ability to effectively take into account the hierarchical structure of the data, which makes it possible to avoid typical shortcomings of conventional flat classifiers reported in study [9]. For example, unlike flat models that ignore taxonomic relationships, the model built integrates these dependences through the PIG weighting criterion. This provides not only increased classification accuracy but also consistency of results between levels. Unlike standard hierarchical approaches, where weighting coefficients are static, dynamic weights allow us to adapt to different depths of the tree, which significantly improves performance.

Unlike the classic local approaches described in paper [8], which build separate classifiers for each level of the hierarchy, the model built uses a global approach with a single classifier with a dynamic weighting criterion Penalized Information Gain (PIG), and it is this criterion that ensures consistency of categorization between levels. This makes it possible to reduce the accumulation of cascading errors that are typical for local methods.

Our model directly solves the following problems:

1. Cascading errors. Dynamic PIG minimizes errors that propagate from the upper levels of the tree to the lower ones, ensuring stable accuracy at all levels. This is evident from the increase in the F1-metric at the Class and Brick levels (Fig. 4, 5), where the built model with dynamic PIG outperforms flat approaches, demonstrating stable accuracy even at deep levels of the hierarchy, where cascading errors usually accumulate. This becomes possible due to the penalization of incorrect splits based on taxonomic information.

2. Uneven data distribution. Our model showed stable results even in cases where the data is concentrated in individual branches of the tree, due to the adaptability of dynamic weights. The uneven distribution of data in our study is manifested in a significant concentration of examples in branches related to the food category, while data on hygiene items are represented by a significantly smaller number of examples. This imbalance occurs at the upper levels of the hierarchy, as food-related categories encompass more subcategories and objects, which creates a challenge for models that focus on consistency of predictions at different levels.

3. Ignoring structure. Unlike flat models, the model built preserves the logic of the hierarchy, which is confirmed by the increase in the F1-metric at all levels.

Our results are appropriate within the framework of clearly defined research conditions:

1. Fixed hierarchy. The model assumes a static tree structure, which limits its application under conditions of dynamic changes in taxonomy.

2. Data preprocessing. The success of the model depends on the quality of text data vectorization (BPE, BERT), which can be difficult for some types of texts.

Ignoring seasonal or other external factors, such as changes in demand for goods depending on the time of year, can lead to a change in the distribution of data (data drift), which affects the accuracy of classification under real conditions. These aspects were not taken into account in the study, which may limit the application of the model in tasks where such factors play a significant role.

The practical significance of the implemented model is its versatility and high accuracy, which makes it suitable for use in many industries. In the field of e-commerce, it can be used for automatic classification of goods, ensuring effective management of product categories, and reducing manual labor costs. In medicine, the model is able to automate the categorization of diagnoses, contributing to accurate grouping of data to support clinical decisions. Other areas, such as text analytics and knowledge management, can significantly benefit from its implementation, especially in the context of complex multi-level data structures. Thus, our model is not only a tool for improving classification accuracy but also a means for solving applied tasks that require preserving hierarchical logic and ensuring high semantic consistency between categories.

Further development of the model built may involve optimizing the weight functions, including studying the influence of various parameters for more effective tuning of dynamic weights, which will allow us to increase the accuracy and consistency of classification. In addition, a promising area is the adaptation of the model to work with dynamic hierarchies that change their taxonomy during operation. Devising approaches that take into account the changing structure of the tree will allow us to expand the capabilities of the model and make it suitable for applications under actual conditions, where hierarchies often evolve. Thus, our model not only solves the current problems of hierarchical classification but also creates the basis for new prospects in research and applied tasks.

7. Conclusions

1. A dataset has been synthesized for training and testing models, which takes into account the structure of a multi-level taxonomy of text data. This allowed us to ensure compliance between the data and the requirements of hierarchical classification, which is important for evaluating models. The dataset included different levels of the hierarchy: Segment, Family, Class, Brick.

2. An algorithm scheme for a hierarchical categorization model was developed, which is based on the dynamic criterion Penalized Information Gain (PIG). The use of dynamically changing weights made it possible to preserve the hierarchical logic of the data and minimize cascading errors, ensuring high semantic consistency of the results.

3. An experimental hierarchical categorization model has been implemented, which was trained and tested on a synthesized dataset from two sources. The model uses semantically enriched text representations obtained using Byte Pair Encoding (BPE) and vectorization via BERT, which made it possible to improve the quality of the input data and ensure their relevance for classification tasks. The implementation of an experimental hierarchical categorization model has confirmed the possibility of integrating the dynamic Penalized Information Gain (PIG) criterion into a decision tree-based model.

4. The testing results showed that the model built with the dynamic PIG criterion demonstrates a noticeable advantage over conventional models, especially at deep classification levels, where the tasks are the most difficult:

– at the Segment level, the highest F1-metric of 0.97 was achieved, which indicates efficiency in categorization of large groups of goods. For comparison, the best F1-metric among other models was 0.95 (Flat_classifier RandomForest), and the worst was 0.92 (Flat_classifier_XGBoost);

– at the Family and Class levels, the model consistently outperformed conventional methods, achieving F1-metrics of 0.87 and 0.78, respectively. For the Family level, the best F1-metric among other models was 0.80 (PerNode RandomForest), and the worst was 0.61 (Flat_classifier_XGBoost). For the Class level, other models showed a maximum of 0.74 (PerNodeRandomForest) and a minimum of 0.54 (Flat_classifier_XGBoost);

– at the smallest classification level – Brick, which is most sensitive to cascading errors, the model with dynamic PIG significantly outperformed its competitors. The achieved F1-metric value of 0.72 (XGBoost) demonstrates the model's ability to accurately account for hierarchical logic even at the deepest levels. For comparison, among other models, the best result was 0.66 (PerNodeRandomForest), and the worst was 0.44 (Flat_classifier_XGBoost).

These results confirm that the proposed model effectively minimizes cascading errors and provides accuracy even at the most complex levels of the hierarchy.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

References

1. Zangari, A., Marcuzzo, M., Rizzo, M., Giudice, L., Albarelli, A., Gasparetto, A. (2024). Hierarchical Text Classification and Its Foundations: A Review of Current Research. Electronics, 13 (7), 1199. https://doi.org/10.3390/electronics13071199
2. Uddin, M. A., Aryal, S., Bouadjenek, M. R., Al-Hawawreh, M., Talukder, M. A. (2024). Hierarchical classification for Intrusion Detection System: Effective at: https://doi.org/10.48550/arXiv.2403.13013
3. Li, J., Wang, H., Song, C., Han, R., Hu, T. (2021). Research on Hierarchical Clustering Undersampling and Random Forest Fusion Classification Method. 2021 IEEE International Conference on Progress in Informatics and Computing (PIC), 53–57. https://doi.org/10.1109/pic53636.2021.9687089
4. Song, J., Zhang, P., Qin, S., Gong, J. (2015). A Method of the Feature Selection in Hierarchical Text Classification Based on the Category Discrimination and Position Information. 2015 International Conference on Industrial Informatics – Computing Technology, Intelligent Technology, Industrial Information Integration, 132–135. https://doi.org/10.1109/iciicii.2015.116

5. Shu, H., Cao, L., Xu, Z., Liu, K. (2009). The Research of Multidimensional Information Decision Mining Based on Information Entropy. 2009 International Forum on Information Technology and Applications, 732–735. https://doi.org/10.1109/ifita.2009.559
6. Doroshenko, A. (2019). Application of Global Optimization Methods to Increase the Accuracy of Classification in the Data Mining Tasks. Computer Modeling and Intelligent Systems, 2353, 98–109. https://doi.org/10.32782/cmis/2353-8
7. Shi, X. (2024). Analysis and Design of the GS1 System Based on ITF-14. International Journal of Computer Science and Information Technology, 3 (3), 206–214. https://doi.org/10.62051/ijcsit.v3n3.21
8. Borges, H. B., Silla, C. N., Nievola, J. C. (2013). An evaluation of global-model hierarchical classification algorithms for hierarchical classification problems with single path of labels. Computers & Mathematics with Applications, 66 (10), 1991–2002. https://doi.org/10.1016/j.camwa.2013.06.027
9. Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. Machine Learning: ECML-98. Berlin, Heidelberg: Springer, 137–142. https://doi.org/10.1007/bfb0026683
10. Silla, C. N., Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 22 (1-2), 31–72. https://doi.org/10.1007/s10618-010-0175-9
11. Basu, S., Bilenko, M., Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/1014052.1014062
12. Sun, A., Lim, E.-P. (2002). Hierarchical text classification and evaluation. Proceedings 2001 IEEE International Conference on Data Mining, 521–528. https://doi.org/10.1109/icdm.2001.989560
13. Read, J., Pfahringer, B., Holmes, G., Frank, E. (2011). Classifier chains for multi-label classification. Machine Learning, 85 (3), 333–359. https://doi.org/10.1007/s10994-011-5256-5
14. Sherif, M. (2020). ProBERT: Product Data Classification with Fine-tuning BERT Model. Available at: https://api.semanticscholar.org/CorpusID:222304838
15. Gao, D., Yang, W., Zhou, H., Wei, Y., Hu, Y., Wang, H. (2020). Deep Hierarchical Classification for Category Prediction in E-commerce System. Proceedings of The 3rd Workshop on E-Commerce and NLP. https://doi.org/10.18653/v1/2020.ecnlp-1.10
16. Shen, D., Ruvini, J.-D., Sarwar, B. (2012). Large-scale item categorization for e-commerce. Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 595–604. https://doi.org/10.1145/2396761.2396838
17. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org/10.18653/v1/n16-1174
18. Gupta, V., Karnick, H., Bansal, A., Jhala, P. (2016). Product Classification in E-Commerce using Distributional Semantics. Proceedings of COLING 2016. Osaka, 536–546. Available at: https://aclanthology.org/C16-1052/
19. Cai, X., Xiao, M., Ning, Z., Zhou, Y. (2023). Resolving the Imbalance Issue in Hierarchical Disciplinary Topic Inference via LLM-based Data Augmentation. 2023 IEEE International Conference on Data Mining Workshops (ICDMW), 1424–1429. https://doi.org/10.1109/icdmw60847.2023.00181
20. Peng, H., Li, J., Wang, S., Wang, L., Gong, Q., Yang, R., Li, B. et al. (2021). Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification. IEEE Transactions on Knowledge and Data Engineering, 33 (6), 2505–2519. https://doi.org/10.1109/tkde.2019.2959991
21. Naik, A., Rangwala, H. (2016). Filter based taxonomy modification for improving hierarchical classification. https://doi.org/10.48550/arXiv.1603.00772
22. Narushynska, O., Teslyuk, V., Doroshenko, A., Arzubov, M. (2024). Data Sorting Influence on Short Text Manual Labeling Quality for Hierarchical Classification. Big Data and Cognitive Computing, 8 (4), 41. https://doi.org/10.3390/bdcc8040041
23. GPC browser. Available at: https://gpc-browser.gs1.org/
24. Jayakody, J., Vidanagama, V., Perera, I., Herath, H. (2023). BERT Layer Weighting Comparision with Short Text Classification. 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), 163–168. https://doi.org/10.1109/iciis58898.2023.10253544
25. Tangirala, S. (2020). Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm. International Journal of Advanced Computer Science and Applications, 11 (2). https://doi.org/10.14569/ijacsa.2020.0110277
26. Araabi, A., Monz, C., i Niculae, V. (2022). How effective is Byte Pair Encoding for out-of-vocabulary words in neural Machine Translation? Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track). Orlando, 117–130. Available at: https://aclanthology.org/2022.amta-research.9/
27. Bao, D., Qin, D., Liang, X., Hong, L. (2021). Short Text Classification Model Based on BERT and Fusion Network. 2021 5th International Conference on Computer Science and Artificial Intelligence, 168–174. https://doi.org/10.1145/3507548.3507574
28. Lima Marinho, T., do Nascimento, D. C., Pimentel, B. A. (2024). Optimization on selecting XGBoost hyperparameters using meta-learning. Expert Systems, 41 (9). https://doi.org/10.1111/exsy.13611
29. Rao, S. X., Egger, P. H., Zhang, C. (2023). Hierarchical classification of research fields in the "web of science" using deep learning. https://doi.org/10.48550/arXiv.2302.00390