CoLeM: A framework for semantic interpretation of Russian-language tables based on contrastive learning

Kirill V. Tobola[1,2], Nikita O. Dorodnykh[1,2],

[1]ISDCT SB RAS, Irkutsk, Russia,
[2]ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia,

Correspondence: kirilltobola@icc.ru

Abstract

Tables are extensively utilized to represent and store data, however, they often lack explicit semantics necessary for machine interpretation of their contents. Semantic table interpretation is essential for integrating structured data with knowledge graphs, yet existing methods face challenges with Russian-language tables due to limited labeled data and linguistic peculiarities. This paper introduces a contrastive learning approach to minimize reliance on manual labeling and enhance the accuracy of column annotation for rare semantic types. The proposed method adapts contrastive learning for tabular data through augmentations and employs a distilled multilingual BERT model trained on the unlabeled RWT corpus (comprising 7.4 million columns). The resulting table representations are incorporated into the RuTaBERT pipeline, reducing computational overhead. Experimental results demonstrate a micro-F1 score of 97% and a macro-F1 score of 92%, surpassing several baseline approaches. These findings emphasize the efficiency of the proposed method in addressing data sparsity and handling unique features of the Russian language. The results further confirm that contrastive learning effectively captures semantic similarities among columns without explicit supervision, which is particularly vital for rare data types.

1 Introduction

Tabular data are one of the key formats for presenting structured information in various domains, ranging from scientific research to business analytics. It is widely used in relational databases, spreadsheets, web resources, and documents, making its processing critically important for automating data analysis. However, tables typically lack explicit semantics necessary for machine interpretation of their content. Therefore, the semantic interpretation of tables, especially in non-English languages, remains a challenging task (Badaro et al., 2023; Liu et al., 2023). The primary challenges are associated with mapping individual table elements (columns, rows, cells) to concepts from knowledge graphs such as DBpedia or Wikidata, as well as handling the structural and linguistic diversity of data.

Russian-language tables pose a particular challenge due to the limited availability of specialized tools and annotated datasets. Most modern methods, particularly those based on pretrained language models like BERT (Deng et al., 2020; Herzig et al., 2020; Yin et al., 2020; Iida et al., 2021; Wang et al., 2021b; Suhara et al., 2022), require vast amounts of labeled data, which are often unavailable or imbalanced for the Russian language. Moreover, existing solutions developed for English do not adapt well to other languages due to differences in tokenization and contextual semantics.

In this paper, we propose a novel approach, called CoLeM, for column type annotation in Russian-language tables based on contrastive learning. This approach effectively leverages unlabeled tabular data to train robust vector representations, reducing the reliance on manual annotation. Our contributions include:

1. Adaptation of contrastive learning for Russian-language tabular data using augmentations such as cell deletion and rearrangement.
2. Utilization of the distilled multilingual model DistilBERT, which balances performance and computational costs.
3. Integration of pre-trained tabular representations into an existing annotation pipeline based on the RuTaBERT (Tobola and Dorodnykh, 2024) framework, demonstrating the flexibility of the approach.
4. Experiments on the large Russian-language dataset, RWT-RuTaBERT, showed that the

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 784–794

proposed approach outperforms certain baseline solutions, confirming its effectiveness under conditions of data sparsity and linguistic specificity.

The paper is organized as follows: Section 2 reviews the current state of research on semantic table interpretation. Section 3 describes the proposed approach for column type annotation in Russian-language tables, including data preparation, model architecture, and training algorithm. Section 4 presents experimental evaluations of the proposed approach's performance. Finally, Section 5 discusses the obtained results and outlines plans for future work.

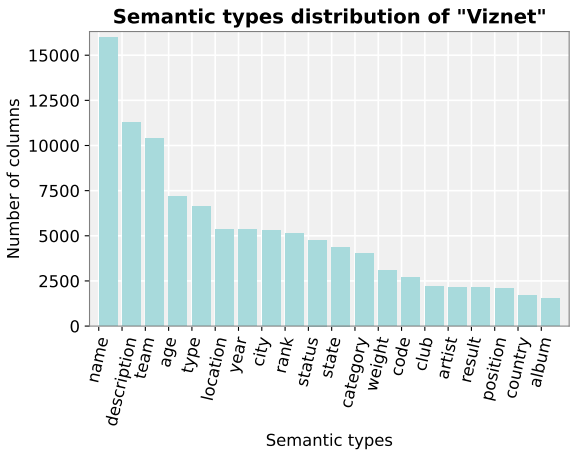

Figure 1: An example of data sparsity issue in the Viznet dataset.

2 Related works

Semantic table interpretation (STI) refers to the process of recognizing and linking tabular data to concepts from a target knowledge graph, ontology, or external vocabulary (e.g., DBpedia, Wikidata, Yago, Freebase, WordNet) (Liu et al., 2023; Zhang and Balog, 2020). One of the core tasks of STI is column type annotation, which involves mapping table columns to semantic types (classes and properties) from the target knowledge graph.

Over the past few years, existing methods and models have leveraged advances in deep machine learning, formulating the column type annotation task as a multi-class classification problem. For instance, (Hulsebos et al., 2019) employed neural networks and various extracted feature groups, such as word and character embeddings, as well as global column statistics. The study by (Zhang et al., 2020) incorporated analysis of local (intra-table) context (adjacent columns relative to the target column), while (Wang et al., 2021a) further added inter-table context to improve predictions. However, particular interest lies in works utilizing pre-trained language models based on the Transformer architecture. Transformer blocks employ an attention mechanism, enabling the model to generate useful contextualized embeddings for structural components of tabular data, such as cells, columns, or rows. Additionally, language models pre-trained on large-scale text corpora can encode semantics from the training text into model parameters, making fine-tuning on specific downstream tasks highly efficient. Examples of such works include models like TURL (Deng et al., 2020), TaPas (Herzig et al., 2020), TaBERT (Yin et al., 2020), TABBIE (Iida et al., 2021), TUTA (Wang et al., 2021b), and Doduo (Suhara et al., 2022).

Existing solutions in this area achieve high performance due to the availability of large labeled training datasets. Specifically, English-language datasets may include hundreds of thousands of labeled columns (e.g., VizNet-Sato (Zhang et al., 2020) $\sim$ 100,000, WikiTables-TURL (Deng et al., 2020) $\sim$ 600,000), while the Russian-language tabular dataset RWT-RuTaBERT contains over 1.4 million columns. Creating such datasets is a labor-intensive process requiring significant time and resources. Moreover, existing table datasets often suffer from data sparsity, manifested in a highly imbalanced distribution of semantic types (known as a "*long-tail distribution*"). For instance, some semantic types correspond to hundreds of thousands of columns, while others are associated with only a few dozen. As a result, models struggle to capture sufficient signals for minority (rare) semantic types (e.g., "*athlete*", "*mountain range*" or "*insurance company*"), even in supervised settings. Figure 1 illustrates this issue with a distribution chart of the 20 most frequent semantic types in the VizNet-Sato dataset. Figure 2 shows the same issue for the RWT-RuTaBERT dataset.

It should also be noted that current methods based on pre-trained language models are not universally applicable. There is a gap between the effectiveness of existing solutions on test cases and their practical applicability, particularly for tables in non-English languages and with varying structural layouts.

To enhance general table understanding and address various tabular tasks, recent works have em-

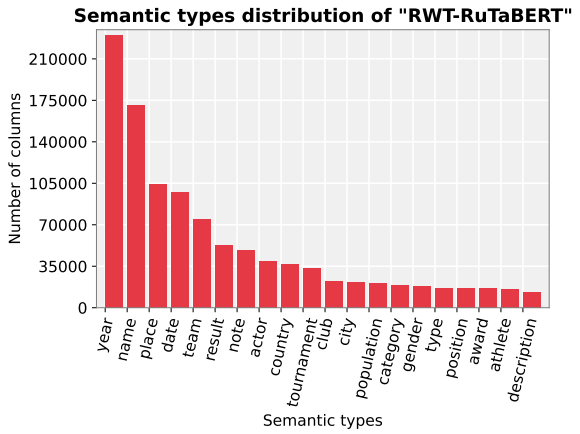

Figure 2: An example of data sparsity issue in the RWT-RuTaBERT dataset.

ployed large language models, which often outperform pre-trained models like BERT. These models are also more robust to unseen examples due to specific effects arising from their scale and training on vast text corpora. Examples include models such as Table-GPT (Li et al., 2024), TableLlama (Zhang et al., 2024), and approaches in (Korini and Bizer, 2024). However, a major drawback of such solutions is their requirement for substantial computational resources, hindering practical use.

To address the aforementioned challenges, we propose the use of self-supervised learning methods, specifically contrastive learning, to derive tabular representations from a large corpus of unlabeled tabular data. These representations can be used for determining relatedness between two tables (via cosine embedding similarity) and for fine-tuning with limited labeled data for specific downstream tasks.

3 Proposed approach

3.1 Problem statement

A table is a two-dimensional data structure composed of rows and columns. Table cells may contain textual data, numerical values, dates, times, etc. Tables can be categorized into three types based on the structure of information:

1. Highly structured (relational database tables);

2. Semi-structured (spreadsheets created in specialized software, e.g., MS Excel);

3. Unstructured (table images in PDF documents).

Tables can also be classified into three main groups based on orientation:

1. Vertical – tables where data is arranged in vertical columns (i.e., top to bottom);

2. Horizontal – tables where data is arranged in horizontal lines (i.e., left to right);

3. Matrix – tables where each entry is indexed by row and column key(s).

This work focuses solely on vertical, highly structured, and semi-structured tables. The formal description of an input table can be represented as:

$$T = \{c_1, ..., c_n\}, c_i = \{v_1, ..., v_m\}, i \in \overline{1, n} \quad (1)$$

where T is a vertical table; c_i is an i-column; v_j is an j-cell of an i-column with $j \in \overline{1, m}$.

Our goal is to predict the column type, i.e., classify each column by its semantic type, such as "*Book*", "*Writer*", "*Genre*" or "*Publication Date*" rather than standard data types like string, integer, or datetime. The proposed approach involves using 170 distinct semantic types derived from selected classes and properties (value properties and object properties) from the general-purpose knowledge graph DBpedia[1]. Only Russian labels for these types (via language tags) were used, as the approach targets the annotation of Russian-language tables. Formally, this task can be described as:

$$P(c_i) \in KG_{st}, KG_{st} = \{st_1, ..., st_{170}\}, \quad (2)$$

where $P(c_i)$ is a predicted semantic type for a i-column; KG_{st} is a set of all semantic types with a cardinality of 170 in this case.

An example of solving the column type annotation task for an input table is shown in Figure 3.

The core idea of the approach is to develop an encoder for robust tabular representations based on contrastive learning, which can then be applied to downstream tasks, specifically semantic annotation of columns in Russian-language tables. The general schema of the proposed approach is presented in Figure 4.

[1] https://www.dbpedia.org/

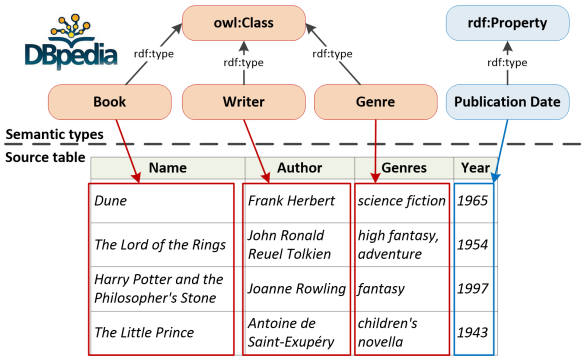

Figure 3: An example of the CTA task.

3.2 Dataset Description

The pre-trained table encoder is trained on a vast amount of tabular data that does not require manual annotation. The large-scale Russian Web Tables (RWT) corpus (Fedorov et al., 2023) is used as the source dataset. This dataset represents a snapshot of tables from the Russian Wikipedia as of September 13, 2021. Key statistics for the RWT corpus are provided in Table 1.

Statistics	Value
Number of tables	1 266 731
Number of columns	7 419 771
Number of cells	99 638 194
Average number of cells per table	81.78
Set size	17 GB
Percentage of almost empty columns	6%
Average number of cells per column	13.42
Percentage of numeric columns	17%

Table 1: Statistics of the RWT table corpus.

During the initial data preprocessing stage, vertical tables were selected from the original RWT corpus. Each column from such a table is represented as a data string using the cell delimiter "«".

Subsequent data cleaning was performed using the following operations:

- Selecting vertical tables.
- Removing empty/sparse columns (<3 cells).
- Filtering extraneous content (parser metadata, Wikipedia links, special characters, such as "@", "&", etc.).

As a result of these cleaning operations, an unlabeled dataset of Russian-language tabular data consisting of 4,656,668 columns was obtained. This preprocessing was automated using a specialized tool, LoReTA.

3.3 Training Algorithm

Contrastive learning is a self-supervised learning technique designed to obtain informative embeddings. It involves maximizing a consistency metric, in our case cosine similarity, between positive pairs (data instances) while minimizing this metric between negative pairs. Contrastive learning enables effective training on unlabeled data corpora.

In this work, we adapt the contrastive learning concept proposed in (Chen et al., 2020) for tabular data. The contrastive learning algorithm for tabular data is illustrated in Figure 5.

The main idea is to construct two augmentations for each column in a batch during training. Column embeddings are generated for the resulting augmentations using an encoder model. Representations of augmentations derived from the same column are considered a positive pair, and our goal is to maximize the cosine similarity metric for this pair. Conversely, representations of augmentations derived from different columns are considered negative pairs, for which the task is to minimize the cosine similarity metric.

3.3.1 Data Augmentation

Data augmentation refers to a technique for artificially increasing the size of a training dataset by applying transformations to the original data. This technique is widely used in scenarios with limited or no labeled data to enhance the model's generalization ability. In contrastive learning, augmentations play a critical role in forming semantically consistent positive pairs.

Common augmentations for tabular data include:

- Random cell deletion.
- Deletion/rearrangement/replacement of tokens in a cell.
- Row sampling (e.g., 50% of rows).
- Cell rearrangement within a table row.
- Column deletion.
- Column rearrangement within a table.

Currently, there is no research identifying the most effective augmentations for forming semantically consistent pairs in the context of tabular data processing. Therefore, in this work, we selected two augmentations deemed most promising: random cell deletion and cell rearrangement within a

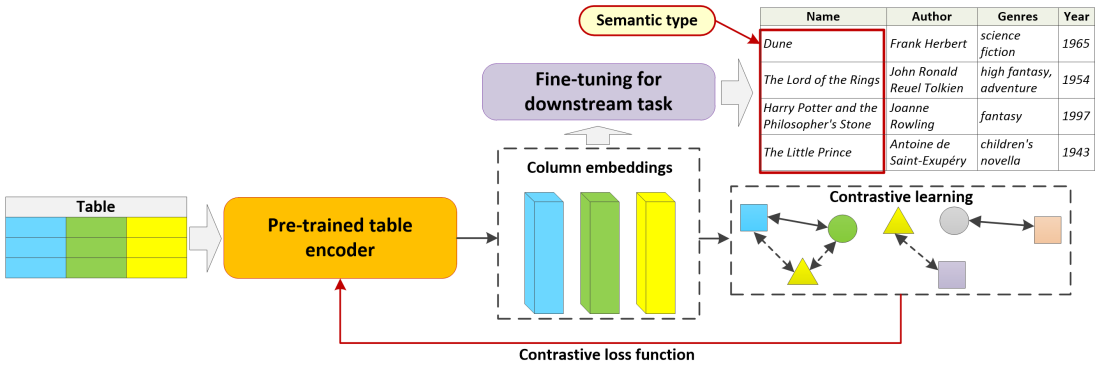

Figure 4: The general scheme of the proposed method integrating self-supervised contrastive pre-training with fine-tuning for downstream tasks (CTA). Key innovations include: (1) Table augmentations (row shuffling, 10% random cell dropping) applied to columns; (2) A distilled multilingual BERT encoder optimized for computational efficiency; (3) A non-linear projection head (128-dim. MLP) generating transformation-invariant latent representations; (4) Seamless integration with the RuTaBERT annotation framework via fine-tuned encoder outputs; This design minimizes GPU memory demands (<10 GB) while enabling 3x larger batch sizes than SOTA equivalents, crucial for scaling to real-world table corpora.

column. For random cell deletion, 10% of all cells in a column are removed.

3.3.2 Contrastive Loss

Contrastive loss functions are widely used in representation learning tasks, as they enable models to better distinguish internal data structures and, consequently, extract more useful representations. A contrastive loss function aims to maximize agreement between positive pairs and minimize agreement between negative pairs in the vector space.

There are several variations of contrastive loss functions. In this work, we adopt the NT-Xent loss (Normalized Temperature Cross-Entropy Loss) used in (Chen et al., 2020), defined as:

$$L = \frac{1}{2N} \times \sum_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)],$$
$$l(i,j) = -\log \frac{exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \times exp(s_{i,k}/\tau)},$$
$$s_{i,j} = \frac{z_i \times z_j}{||z_i|| \times ||z_j||} \quad (3)$$

where $1_{[k \neq i]}$ is 1 if $k \neq i$, otherwise 0; τ is the temperature parameter; and s is cosine similarity.

3.4 Model Architecture

Currently, Transformer-based models are central to natural language processing tasks. These models are versatile tools for text processing due to their ability to capture contextual dependencies between words in sequences and to train on unlabeled or partially labeled data. They achieve this efficiently through high parallelism, making them preferable for training on large datasets.

According to (Chen et al., 2020), two critical hyperparameters in contrastive learning are batch size and the number of epochs. Larger batch sizes and more epochs result in more representative embeddings, leading to better performance on downstream tasks during fine-tuning.

Based on this, the distilled multilingual BERT model[2] was chosen as the base encoder. This model was trained on Wikipedia articles in 104 different languages. Unlike the base version[3], it consists of only 6 layers (half the number of the base version) and 12 attention heads. It has 134 million parameters (compared to 177 million in the base version).

Model distillation is a technique in machine learning where knowledge is transferred from a more complex model (teacher) to a more compact one (student) while maintaining prediction quality.

This technique, combined with reducing the tokenizer's maximum sequence length to 256 tokens, enabled training with a batch size of 800, which is 25 times larger than that of a comparable state-of-the-art English-language solution (Miao and Wang, 2023).

Research in (Chen et al., 2020) explored the use of projecting the encoder's output layer into a la-

[2] https://huggingface.co/distilbert/distilbert-base-multilingual-cased

[3] https://huggingface.co/google-bert/bert-base-multilingual-cased

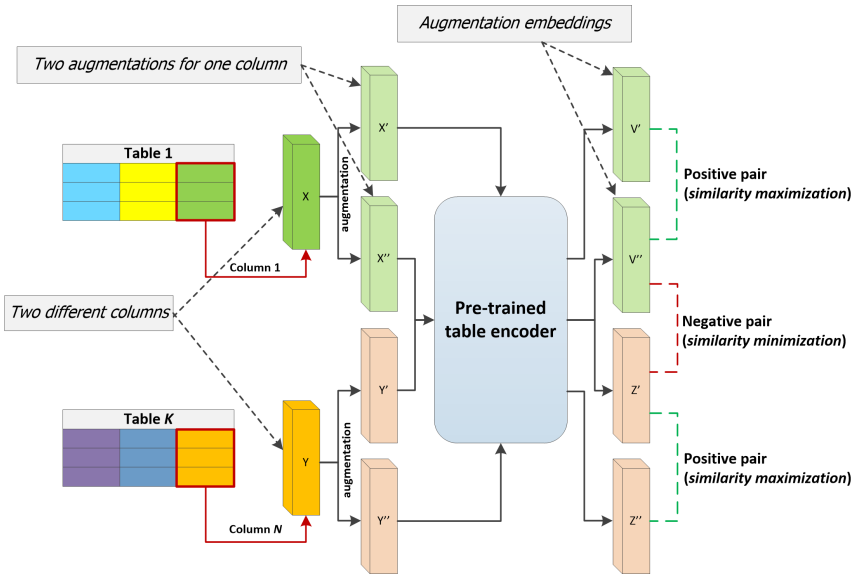

Figure 5: Contrastive learning algorithm for tabular data. Algorithmic workflow demonstrating CoLeM's core innovation: Self-supervised similarity learning via the NT-Xent loss optimization. For each target column, two augmented views are generated. Positive pairs (same column, different augmentations) are embedded closer in latent space, while negatives (all other columns in the batch) are repelled. The temperature-scaled cross-entropy loss ($\tau = 0.1$) forces discriminative feature extraction without manual labels. Crucially, this algorithm captures linguistic and structural patterns specific to Russian tables validated by 15.1% average Macro F1 gain over RuTaBERT on rare types (see Table 4) without labeling dependence.

tent space for calculating the contrastive loss. Results indicate that applying a non-linear projection during training positively impacts representation quality. Thus, in this work, a two-layer perceptron (MLP) is used after the encoder's output layer to project into a 128-dimensional latent space where the contrastive loss is computed using the aforementioned formula.

4 Experimental Evaluation and Discussion

All experiments were conducted on the compute cluster "Akademik V.M. Matrosov"[4] on the basis of the Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences (ISDCT SB RAS). The cluster configuration includes two 16-core Intel Xeon Gold 6326 "Ice Lake" 2.9 GHz processors, four NVIDIA A100 80 GB PCIe GPUs, and 2 TB of DDR4-3200 RAM.

4.1 Contrastive Learning Setup

The approach was implemented in Python using the PyTorch and Transformers libraries. The AdamW optimizer ($lr = 5 \times 10^{-5}$, $eps = 10^{-6}$) was chosen for gradient descent. To accelerate convergence, cosine annealing was applied to dynamically reduce the learning rate. The temperature parameter, a hyperparameter of the contrastive loss function, was set to 0.1, as this value was found to be optimal in (Chen et al., 2020). Under these settings, the pre-trained encoder model was trained for 100 epochs on 4 NVIDIA A100 GPUs using the Distributed-Data-Parallel technology of the PyTorch framework. Training lasted 9 days, 9 hours, and 53 minutes. GPU memory consumption amounted to 290 GB. The source code for CoLeM is published at github[5].

4.2 Column Type Annotation Setup

In this work, column type annotation task was selected as the downstream task. Previously, the RuTaBERT framework was proposed for this task, based on fine-tuning a pre-trained multilingual BERT model using the specially prepared RWT-RuTaBERT dataset. This dataset contains approximately 1.56 million labeled columns. The core idea is to utilize the existing pipeline of this framework, replacing the standard BERT model with a specialized pre-trained table encoder. The RWT-RuTaBERT dataset, with all standard settings, was used for training. The RWT-RuTaBERT dataset

[4] https://hpc.icc.ru

[5] https://github.com/YRL-AIDA/CoLeM

has a fixed split into train and test subsets. The test subset comprises over 115,000 columns (across more than 55,000 tables, with an average of 2.09 columns per table). All performance measurements were conducted on this fixed test subset. The validation set comprised 5% of the total training subset. The technique of neighboring column serialization was used to decompose column values into token sequences.

According to (Chen et al., 2020), the projection layer is trained to be invariant to data transformations, potentially losing information useful for downstream tasks. Therefore, for further fine-tuning of the table encoder, the output from the first linear layer of the projection with a LeakyReLU activation function was used. Standard training settings defined in the RuTaBERT framework were applied. The model was fine-tuned for 30 epochs with a batch size of 32 on the RWT-RuTaBERT dataset using 2 NVIDIA A100 GPUs. Training lasted 2 days, 20 hours, and 15 minutes, with GPU memory consumption of 9.9 GB. Additionally, a model with a batch size of 256 was trained with all other hyperparameters unchanged. Under these settings, training took 4 days, 3 hours, and 1 minute, with GPU memory consumption of 52 GB. Pre-trained versions of the RuTaBERT model, utilizing CoLeM as the base encoder (with batch sizes of 32[6] and 256[7]), are available at huggingface.

4.3 Evaluation Metrics

The primary metrics for evaluating the performance of the proposed method are averaged F1 scores, as the task involves multi-class classification. Specifically, Micro F1, Macro F1, and Weighted F1 are used due to the imbalance in the RWT-RuTaBERT dataset.

4.4 Results and Discussion

The results of the experimental evaluation are presented in Table 2. A comparison of the performance of the proposed approach with several baseline solutions is provided.

Firstly, a pre-trained language model, RuBERT (Kuratov and Arkhipov, 2019), which specializes in processing the Russian language, was selected. One of the transfer learning techniques was applied, where the weights of the encoder layers remained unchanged during training. Thus, during fine-tuning of RuBERT on the RWT-RuTaBERT dataset, only the parameters of the classification layer were adjusted.

Model	micro F1	macro F1	weighted F1
Doduo	0.140	0.040	N/A
RuBERT-ft	0.610	0.410	0.590
Doduo-ft	0.962	0.890	0.960
RuTaBERT	0.964	0.900	0.963
CoLeM-bs32	*0.969*	*0.910*	*0.969*
CoLeM-bs256	**0.974**	**0.924**	**0.974**

Table 2: Results of experimental evaluation on the RWT-RuTaBERT dataset and comparison with baselines. "N/A" denotes not applicable in their original framework.

Secondly, the Doduo (Suhara et al., 2022) framework was chosen. Doduo is a state-of-the-art (SOTA) model for column type annotation in English tables, trained on the Viznet-Sato dataset. It uses a pre-trained BERT model as the base encoder for tabular representations and proposes a table serialization method that predicts semantic types for all columns in a single forward pass. In this case, transfer learning was also applied by freezing the transformer layers and fine-tuning only the final linear classifier layer. Additionally, a full fine-tuning of the multilingual BERT model was performed following the Doduo approach on the RWT-RuTaBERT dataset (Doduo-ft). Unlike Doduo, CoLeM is a versatile encoder for tabular representations, designed for integration into existing solutions for semantic table interpretation. Trained on a corpus of tables from Russian Wikipedia, it is primarily oriented toward the Russian language. However, CoLeM leverages a multilingual BERT model as its base, suggesting potential applicability to other languages, which will be explored in future research.

Thirdly, the original RuTaBERT approach was considered. RuTaBERT adapts Doduo's concepts for the Russian language, utilizing local table context (neighboring columns) for column annotation. It introduces a new table serialization approach, predicting the semantic type of a single target column per forward pass, with other columns serving as context. On Russian tables, RuTaBERT slightly outperforms Doduo in micro-F1 (by less than 1%) and shows a 1% improvement in macro-F1.

The obtained evaluation results demonstrated

[6] https://huggingface.co/sti-team/colem-rutabert-32bs

[7] https://huggingface.co/sti-team/colem-rutabert-256bs

that the proposed approach outperformed all baseline solutions in both training configurations (batch sizes of 32 and 256). Specifically, the experiment showed that while the RuBERT model is tailored for processing the Russian language, it is not directly suited for tabular tasks, which proved challenging for this model. Consequently, existing Russian-language models cannot be effectively applied to the column type annotation task.

The Doduo model, trained using transfer learning techniques, exhibited relatively low evaluation results. This is attributed to the fact that the model was trained on tabular data exclusively in English. Notably, the tokenizer of this model lacks sufficient Russian-language tokens. As a result, it can be concluded that a model trained on English data cannot be directly applied to another language, such as Russian, without modifying the base encoder to accommodate the target language.

Meanwhile, the fine-tuned multilingual encoder of the Doduo framework and the RuTaBERT approach demonstrated nearly comparable results in terms of evaluation metrics. However, it can be observed that the use of a pre-trained tabular encoder based on contrastive learning positively impacts the performance. With a smaller model and identical settings, the proposed approach achieved results equivalent to those of the classical RuTaBERT model or the fine-tuned Doduo. Additionally, the model consumes approximately three times less GPU memory during training, requiring less than 10 GB (with a batch size of 32, consistent across all three models), which enables training on a standard home computer. Furthermore, with a larger batch size (e.g., 256), the proposed approach achieved a performance gain of 1.5% compared to the classical RuTaBERT model and nearly 3% compared to the fine-tuned Doduo. The experimental results highlight the potential of our approach for semantic annotation of Russian-language tables.

To further evaluate CoLeM's performance, we conducted a statistical analysis on three aspects:

1) Datatype groups: The original test set, comprising 115,448 columns, was divided into 6 groups by mapping existing semantic types to a set of 6 general categories (data types). All columns from the original test set were utilized. **Numeric** includes 4,592 columns with semantic types such as distance, population, area, weight, depth, age, etc. **Date** includes 29,473 columns with semantic types such as year, date, day, period, duration. **Person** includes 7,504 columns with semantic types such as actor, screenwriter, judge, producer, footballer, character, chess player, etc. **Links** includes 103 columns with semantic types such as link, website. **Long Text** includes 5,850 columns with semantic types such as address, document, annotation, location, description, note, etc. **Short Text** includes 67,926 columns with semantic types such as car, race, genre, animal, team, nationality, etc.

CoLeM, similar to other language models, may encounter challenges with numeric values as it processes all cells as strings. However, the overall performance on numeric data suggests that transformers possess a partial capability to analyze numerical sequences. Table 3 summarizes the Micro F1 score and distribution for each datatype group.

Data type	F1 (CoLeM)	F1 (RuTaBERT)
Datetime	**0.948**	0.941
Long text	0.858	**0.885**
Numeric	**0.760**	0.749
Person	**0.716**	0.692
Short text	**0.932**	0.926
Links	0.611	**0.699**

Table 3: Results of model evaluation (Micro F1) for 6 datatype groups. Columns were classified into basic 5 groups: Datetime (dates/times), Numeric (measurements), Links (including URLs), Short Text (< 4 tokens), and Long Text (≥ 4 tokens). Persons data type was added for role-based entries (e.g., *"employer"*).

2) Rare semantic types: Performance evaluations were also conducted for the 15 least frequently occurring semantic types. For comparison, checkpoints of the CoLeM-bs32 and RuTaBERT models, which achieved the highest macro F1 score on the training set, were used. The results are presented in Table 4.

The results demonstrate that, due to the robust tabular representations obtained, the CoLeM model significantly outperforms the existing state-of-the-art (SOTA) Russian-language solution, RuTaBERT, in terms of evaluation metrics for infrequently occurring semantic types.

3) Model convergence: To evaluate the convergence of the CoLeM model, experiments were conducted for checkpoints of CoLeM-bs32 and RuTaBERT models trained for 10 and 30 epochs. The performance results are summarized in Table 5.

It can be observed that the CoLeM model converges faster than the RuTaBERT model and has 1-3% better performance. This allows us to use a smaller number of epochs in training stage,

while obtaining comparable or even superior performance to the RuTaBERT model.

Broader applicability and generalizability

The proposed CoLeM framework presents a significant advancement in semantic table interpretation for Russian-language tables by leveraging contrastive learning and distilled multilingual BERT model. Its core innovation is to minimize dependence on labeled data and efficiently handle rare semantic types, which demonstrates remarkable potential for adaptation to low-resource languages. To deploy CoLeM beyond Russian, the following minimal adjustments are needed:

1. *Corpus Construction*: Replace RWT with locally sourced unlabeled tables (e.g., from government portals, local-language Wikipedia). The cleaning pipeline (cell value filtering, metadata removal) remains unchanged. For languages with non-Latin languages (e.g., Arabic, Thai), ensure Unicode normalization during preprocessing.

2. *Tokenizer Specialization*: While multilingual BERT's tokenizer covers major languages, extremely low-resource languages (e.g., the varieties of Finno-Ugric languages) may require extending the vocabulary via subword sampling on target-language corpora.

3. *Knowledge Graph Alignment*: Replace DBpedia with localized knowledge graphs (e.g., BabelNet for cross-lingual types, or domain-specific ontologies). At the same time, the 170-type schema can be reused or expanded.

5 Conclusion

This study proposes an approach for semantic annotation of columns in Russian-language tables based on contrastive learning. The experimental results demonstrate that the approach mitigates the dependency on large volumes of labeled data by leveraging self-supervised learning on unlabeled tables. Moreover, it outperforms existing baseline solutions (Doduo and RuTaBERT) in terms of evaluation metrics, particularly for rare semantic types. The approach also ensures computational efficiency through the use of a distilled model and optimized batch sizes, reducing memory requirements by 60% compared to analogous methods.

The results of the experimental evaluation confirm the effectiveness of the proposed solution. In the future, as part of a research project with the Ivannikov Institute for System Programming of the Russian Academy of Sciences (ISP RAS), it is planned to integrate these results into a specialized table processor within the Talisman platform[8]. Additionally, we plan to investigate the potential application of the proposed column encoding method to other types of tables (horizontal and matrix-based). We will also address specific challenges that arise when working with these different table structures. Further investigation will also focus on the use of new data augmentations to enhance the robustness of tabular representations.

Overall, the proposed approach opens up opportunities for the development of universal systems for semantic interpretation of tables, which is relevant for tasks involving the integration of structured and semi-structured information, as well as business analytics.

Limitations

CoLeM shows strong performance with Russian-language tables and potential for broader language application, yet it faces limitations. Firstly, its structural augmentations (cell deletion/rearrangement) are suited to vertical layouts, leaving complex matrix or horizontal tables (e.g., in financial reports) unaddressed. Secondly, the multilingual DistilBERT tokenizer, despite supporting 104 languages, struggles with agglutinative languages (e.g., Finnish, Turkish) and scripts needing unique segmentation (e.g., Khmer, Amharic), requiring tailored tokenization. Thirdly, reliance on DBpedia as a semantic schema overlooks culture-specific concepts vital for low-resource languages, complicating local ontology integration. These challenges underscore the need for hybrid augmentations, script-adaptive tokenization, and adaptable knowledge graph integration in future research.

Acknowledgments

This work was supported by a grant, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4G0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated June 20, 2025 No. 139-15-2025-011.

[8] http://talisman.ispras.ru

References

Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2023. Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 11:227–249.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, pages 1597–1607.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. *Proceedings of the VLDB Endowment*, 14(3):307–319.

Platon E. Fedorov, Alexey V. Mironov, and George A. Chernishev. 2023. Russian web tables: A public corpus of web tables for russian language based on wikipedia. *Lobachevskii Journal of Mathematics*, 44:111–122.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'2020)*, pages 4320–4333.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, pages 1500–1508.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. Tabbie: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456.

Keti Korini and Christian Bizer. 2024. Column property annotation using large language models. In *Proceedings of the Semantic Web: ESWC 2024 Satellite Events*, pages 61–70.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle R. Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data*, 2(3):1–28.

Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, 76:100761.

Zhengjie Miao and Jin Wang. 2023. Watchog: A lightweight contrastive learning based framework for column annotation. *Proceedings of the ACM on Management of Data*, 1(3):1–24.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD'22)*, pages 1493–1503.

Kirill V Tobola and Nikita O Dorodnykh. 2024. Semantic annotation of russian-language tables based on a pre-trained language model. In *2024 Ivannikov Memorial Workshop (IVMEM)*, pages 62–68. IEEE.

Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. 2021a. Tcn: Table convolutional network for web table interpretation. In *Proceedings of the Web Conference (WWW'21)*, pages 4020–4032.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021b. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD'21)*, pages 1780–1790.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'2020)*, pages 8413–8426.

Dan Zhang, Madelon Hulsebos, Yoshihiko Suhara, Çağatay Demiralp, Jinfeng Li, and Wang-Chiew Tan. 2020. Sato: Contextual semantic type detection in tables. *Proceedings of the VLDB Endowment*, 13(11):1835–1848.

Shuo Zhang and Krisztian Balog. 2020. Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(2):1–35.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. Tablellama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6024–6044.

A Appendix: Evaluation for 15 least frequently occurring semantic types

Semantic type	Number of samples (test subset)	F1 (RuTaBERT)	F1 (CoLeM-bs32)
camera	102 (4)	0.250	**0.750**
employer	101 (10)	0.899	**1.000**
device	101 (8)	0.625	**0.875**
animal	93 (7)	0.857	**1.000**
magazine	93 (9)	0.440	0.440
continent	92 (8)	0.625	**0.750**
novel	89 (11)	0.818	**0.909**
law	89 (9)	1.000	1.000
wrestler	88 (5)	0.400	**0.600**
college	87 (5)	0.000	**0.200**
museum	86 (4)	0.500	**0.750**
firm	85 (6)	0.333	0.333
prefecture	83 (10)	0.600	**0.699**
road	83 (6)	0.500	**0.666**
quote	76 (7)	0.857	**1.000**

Table 4: Performance evaluations for the 15 rarest semantic types compared CoLeM-bs32 and RuTaBERT (best training-set Macro F1 checkpoints). The results show CoLeM's tabular representations outperform RuTaBERT (Russian SOTA) on infrequent types and capture linguistic and structural patterns specific to Russian tables (15.1% average Macro F1 gain over RuTaBERT).

B Appendix: Model evaluation after 10 and 30 training epochs

Table 5: Results of model evaluation after 10 and 30 training epochs. Experiments on CoLeM-bs32 and RuTaBERT show CoLeM converges faster with 1-3% higher performance, enabling fewer training epochs while matching/exceeding RuTaBERT results.

Model	Micro F1	Macro F1	Weighted F1
RuTaBERT (10 epochs)	0.952	0.856	0.952
CoLeM-bs32 (10 epochs)	**0.966**	**0.888**	**0.966**
RuTaBERT (30 epochs)	0.964(+0.012)	0.904(+0.048)	0.963(+0.011)
CoLeM-bs32 (30 epochs)	**0.969**(+0.003)	**0.910**(+0.022)	**0.969**(+0.003)