ISSN 0005-1055, Automatic Documentation and Mathematical Linguistics, 2025, Vol. 59, Suppl. 6, pp. S571–S590. © Allerton Press, Inc., 2025.
Russian Text © The Author(s), 2025, published in Elektronnye Biblioteki, 2025, Vol. 28, No. 6, pp. 1481–1519.

Automatic and Semiautomatic Methods for Domain Knowledge-Graph Construction and Ontology Expansion

A. P. Khalov[a, *] **(ORCID: 0009-0005-4584-8245) and O. M. Ataeva**[a, b, **] **(ORCID: 0000-0003-0367-5575)**

[a] *Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Moscow oblast, Russia*

[b] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*

**e-mail: khalov.a@phystech.edu*

***e-mail: oataeva@frccsc.ru*

Received November 11, 2025

Abstract—We present a combined pipeline for knowledge-graph construction and ontology expansion. This approach creates a BIO-tagged corpus via fully automatic LLM-based pseudoannotation and introduces dedicated UNK reserve categories to capture previously unseen classes and relations. A specialized NER/RE model is trained on a 3-million-token dataset with 92 labels. This model exhibits a conservative quality profile—high precision with moderate recall—suited for safe graph enrichment: integrating the extracted facts expands the graph to ~0.98 million triples, while the expansion ratio (total inferred facts to explicit triples) increases from 2.65 to 3.52, with logical consistency preserved. UNK label pools are converted into stable synsets, enabling semiautomatic ontology expansion; 12 new classes derived from unstructured texts were added. We also demonstrate practical value for querying and analytics using an LLM + SPARQL setup.

Keywords: ontology, DOLCE, knowledge graph, NER, BIO tagging, RDF/OWL, SPARQL

DOI: 10.3103/S0005105525701547

INTRODUCTION

Ontologies and knowledge graphs are classic tools for integrating heterogeneous data and supporting decision-making in business processes of managing (IT services) data flows (IT domain) in IT Service Management (ITSM). Previously [1], a unified IT domain ontology was formed by extending the top-level ontology DOLCE [2] with a domain-specific ontology ITSMO [3] that describes the concepts of the ITIL library. In practice, the value of a knowledge graph is determined by the speed and accuracy of replenishment with facts from texts (such as IT domain data, which is managed by IT services) while simultaneously being able to adapt a schema (ITSMO ontology) to the changing thesaurus of the domain.

For the dynamic construction and replenishment of knowledge graphs, as well as for the expansion of the ontology, it is necessary to have specialized models for the extraction of named entities and relations (hereinafter NER/RE models), trained on data corpora, the objects in which are marked in accordance with the ontology. However, at the time that this study was conducted, there were no open data sets for ITSM labeled in accordance with the ontology from [1] (hereinafter ontology). At the same time, there is access to large volumes of text data in untagged form. Manual or semiautomated labeling for training the NER/RE model is a labor-intensive and expensive process.

In this paper, a method for fully automatic labeling of a training data corpus is investigated. One well-known markup method for NER/RE tasks is the BIO format (Beginning-Inside-Outside), which contains three types of labels: to denote the first token of an entity (B); any subsequent token of an entity (I); the background token (O). Another objective of the research is to create a method for expanding ontology, i.e., detecting entities and relationships in texts that do not correspond to the existing ontology but can be a source of valuable information for its expansion.

This study is divided into three stages:

(1) formation of a BIO corpus of texts with pseudo-markup using large language models (LLMs), where we introduce three sets of labels, as well as special UNK-labels (from English UNKnown) in each of the defined sets for marking words that have no correspondence in the current ontology;

(2) training an encoder model from the BERT family for the task of extracting named entities and relations with correctness checking at the level of label types;

(3) automatic knowledge graph enrichment and semiautomatic ontology expansion through expert validation of clusters of vector representations (embeddings) of words that have been marked as UNK.

The subject area under study is characterized by a high dynamics of terminology changes and a large volume of text artifacts (e.g., customer requests, incidents, and configuration changes). These characteristics define the need for a method that combines cost-effective and scalable data labeling with automatic graph enrichment and controlled ontology expansion while maintaining logical consistency.

This article has the following structure: Section 1 presents an overview of the subject area and related works, Section 2 describes the data corpus and pseudolabeling techniques, Section 3 presents the architecture of the model, Section 4 provides the SQL-RDF-NER-RDF pipeline, Section 5 gives the main results, Section 6 indicates the semiautomatic ontology extension, and the Conclusion follows.

The appendices contain the text of the instructions for LLM (the prompt) at the stage of testing the candidate models (Appendix 1), the prompt for the final pseudolabeling of the target data corpus (Appendix 2), the results of the experiment with testing different models for BIO benchmarks (Appendix 3), the distribution of labels after applying the pseudolabeling method (Appendix 4), examples of stable semantic sets (synsets, abbreviated from synonym set), and their transformations into ontology objects (Appendix 5).

1. OVERVIEW OF THE SUBJECT AREA AND RELATED WORKS

Ontology is a formal specification of a conceptualization [4, 5]; in a more applied interpretation, it is a consistent representation of a subject area that is shared by a community [6, 7]. In information extraction and the construction of knowledge graphs, ontology acts as a data schema: entities and relationships between them are expressed through classes and properties with explicit logical constraints (axioms), which ensures interpretability, logical inference, and consistent checking of the result.

Our work utilizes a unified ontology: the top level is defined by the well-known DOLCE ontology, using abstract categories (object, process, event), which guarantees ontological coherence; the application layer is formed by the ITSMO ontology, encompassing entities and relationships drawn from the well-known ITIL methodology (e.g., incident, service, change) and their interrelations. This unification yields a single semantically consistent structure that is both conceptually rigorous and practically relevant for automatic text annotation and graph enrichment. The role of ontology in this study is as follows: (i) it is a framework for mapping extracted facts into RDF triples, and (ii) it is a living schema that should be extended as new terms and relationships appear in the incoming data.

1.1. Ontologies and Knowledge Graphs for the IT Domain

Work on ontological modeling covers both high-level ontologies (SUMO, BFO, and DOLCE [6]) and applied domain ontologies, for example ITSMO for ITIL processes [8]. The merger of ITSMO with DOLCE follows the recommended inheritance strategy and avoids the manual design of large parts of the schema [9]. Practices for constructing and operating corporate knowledge graphs are systematized in guidelines [7, 10], where emphasis is placed on the role of ontologies and logical inference for data quality control and consistency.

1.2. Joint Extraction of Named Entities and Relationships

Automatic graph enrichment often comes down to extracting entities and relationships from text. Classical cascades of sequential extraction of entities and relations from texts (NER + RE) are gradually giving way to unified solutions.

An early end-to-end approach was the architecture proposed in [12]: two parallel neural networks based on a common BiLSTM encoder are used to simultaneously extract entities and relations. This approach ensures that the model tends to make consistent predictions, as inconsistent labels (when a token is simultaneously labeled as both an entity and a relation) lead to an increase in the overall loss function of the model.

Further work strengthened the logical constraints in learning: a semantic loss function taking into account a priori symbolic knowledge [13] allowed for an increase of the coherence of predictions; interactions between NER and RE tasks in a single architecture were investigated [14]; and models for hierarchical multiclass classification were proposed that guarantee the consistency of outputs due to the network structure [15]. In parallel, lines of structurally consistent approaches to information extraction (IE) were developed: NER was formulated as dependency analysis [16], and in [17], the authors proposed generative universal IE models. The effectiveness of such approaches increases with additional training (domain-adaptive pretraining) of models using the method of modeling natural language with masking of part of the tokens of the input sequence (masked language modeling, MLM) [18]. In the same work, it was shown that a reduction in the loss function (and the associated MLM-specific perplexity metric) has a direct correlation with an improvement in the performance of the model on applied problems.

1.3. Pseudolabeling and Formation of a Dictionary of BIO-labels

Another direction, one that is close to our task, is automatic data labeling (hereinafter pseudolabeling) using generative pretrained transformer models (GPT). The authors of the GPT-3 model demonstrated the model's ability to perform arbitrary tasks with minimal tuning (one-shot/few-shot learning) [19]. The work [20] demonstrated that open LLMs (e.g., LLaMA, Falcon) can effectively annotate texts,

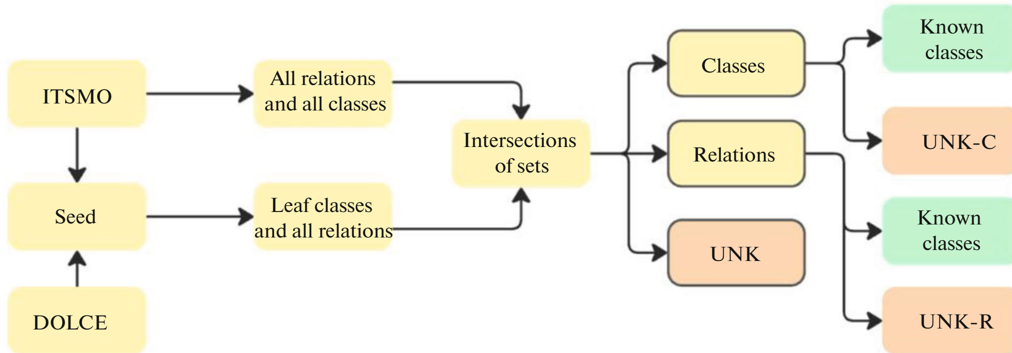

Fig. 1. Scheme for forming dictionaries of class labels.

approaching the quality of the GPT-4 model. At the same time, some studies point to the risks of using generative artificial intelligence (AI) for such purposes [21], primarily the risk of transferring hallucinations and biases into the final dataset. It is worth noting that the cited works focus on the tasks of annotating texts in free form, i.e., they do not consider specialized approaches to annotation at the token sequence level, as required by the BIO markup format for the NER/RE task. In our work, we build on these ideas and apply GPT models to fully automatic BIO tagging of a specialized Russian-language corpus of IT support requests.

2. BIO PSEUDOMARKUP

The experiment used a domain-specific corpus containing over 660000 text queries to the IT support service (the dataset is predominantly in Russian, and about 20% of the texts are in English). Domain concepts are formed on the basis of ontology.

From the ontology, we obtained a nomenclature of 92 labels, which, in addition to classes and relations, included the special labels UNK-x (separately for classes, relations/properties, and types), as well as the auxiliary relation I_A, denoting the “is-a” relationship.

The problem being considered is of transformation of source texts into a dataset, which is then (see Sec. 4) used to train the NER/RE model. For each text, three parallel sequences of BIO tags must be generated. The first sequence specifies the class of the entity (including a class that is not included in the ontology), the second specifies the class of the relationship (including a relationship not included in the ontology), and the third, more abstract, sequence specifies the type: the token belongs to a class (CLA), a relationship (REL), or an unrecognized type (UNK); alternatively, it does not carry a semantic load (O).

Special UNK family tags are found in all three types: UNC-C—for unknown classes, UNK-R—for unknown relations, and UNK—to indicate an undefined type. UNK labels are necessary for the task at hand for two reasons: first, they allow the model to correctly process an unknown entity without forcing it to be assigned to the closest known class or relation; second, they form a separate dataset. After being combined into words, the tokens that were labeled UNK-C and UNK-R form a set of lexical units that can be clustered after vectorization (creation of embeddings). Clusters form a set of synsets that allow us to identify candidates for new classes and relations/properties to extend the ontology. Figure 1 shows a diagram of the division of the composition of the ontology into subsets of classes and relations. The combined ontology is indicated in the figure as Seed.

The pseudomarkup process is subject to a limitation to determine the maximum possible amount of markup, which necessitates careful optimization of both the size of the corpus and the length of instructions (prompts) passed to the language model.

2.1. Methodology for Selecting a Model for Pseudolabeling

To evaluate the ability of language models for BIO-labeling, four well-known corpora for NER tasks, manually labeled (hereinafter referred to as benchmarks), were selected:

• CoNLL 2003 [22] contains 20000 newspaper sentences marked with nine abbreviation marks and is considered a classic English-language standard.

• WikiAnn [23] is presented in two versions: English (en) and Russian (ru); each includes approximately 40 thousand objects and abbreviation labels; this allows the multilingualism of models to be tested.

• WNUT building 17 [24] is significantly smaller in volume (about 5.5 thousand objects); it differs in that here the labels are independent concepts: *corporation, creative work*, etc., unlike the others, where labels are implemented as three-letter indices.

In all, 12 LLMs were tested: 8 proprietary (models from OpenAI and Anthropic) and 4 open models (Llama3, Mistral, and DeepSeek). The prompts contain instructions and examples of markup [25] (see Appendix 1). Each model was given 2000 tagging tasks, after which the optimal model for the pseudot-

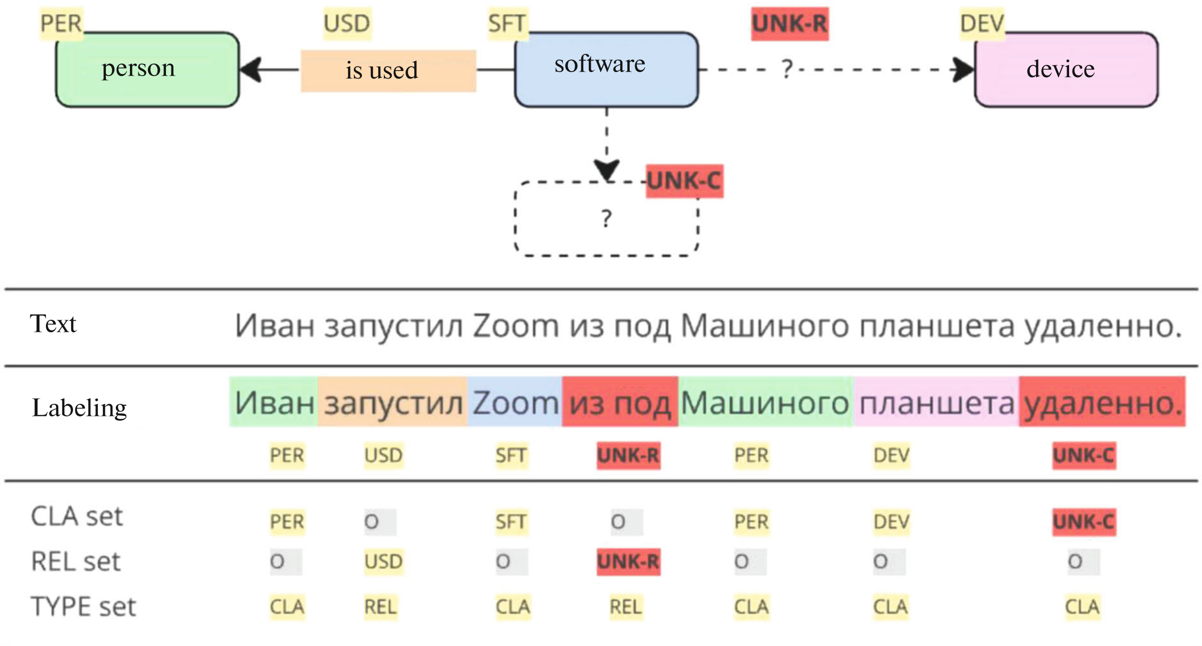

Fig. 2. Illustration of markup using a miniontology as an example.

agup of the target text corpus was determined, showing the maximum metric values.

When the models were tested, two types of F1 metrics were used as evaluation criteria. Token F1 considers each token individually: the score may be high if the model correctly predicts the labels of most tokens in a sequence. Seqeval F1 is a more stringent criterion; for it, a prediction is considered correct only if the model correctly identifies both the boundaries and the type of the entire entity. The difference between Token F1 and Seqeval F1 indicates the ability of the model to maintain the integrity of the label sequence at the entire entity level.

2.2. Methodology of the Pseudomarkup of Target Text Corpora

During the pseudolabeling process, at each iteration, the language model could use all 92 labels simultaneously. In the post-processing stage, the sequence labels were divided into two disjoint sets of labels (classes and relations) and a third derived set that captures only the token type: class, relation, unknown (UNK), or token without semantic load (O). This ensures that the three BIO sequences are mutually exclusive and logically consistent.

The prompt consists of four parts: instructions, a markup example, markup text, and a label dictionary with a brief description of the semantics of each label (see Appendix 2).

2.3. Illustration of the Markup

The illustration of the processing pipeline is presented as a conditional miniontology (in Fig. 2), the entities of which are transformed into a set of labels and then applied to the sentence and decomposed into three parallel BIO-sequences: classes (CLA set), relations (REL set) and token types (TYPE set), as shown in the figure below. Elements that are not present in the miniontology are marked as UNK-C or UNK-R.

For demonstration (Fig. 2) a primitive ontology with three classes (PER, person; SFT, software; DEV, device) and one relation (USD, used) was used. The sentence also contains an unknown relation with the label UNK-R and an unknown class with the label UNK-C, which are not in the ontology but are explicitly expressed in the text. This allows the system to flag candidates for subsequent ontology expansion.

For example, the word *"remotely"* is marked as UNK-C because the concepts *"method"/"mode"* do not have a suitable class in the ontology; the phrase *from under* is marked as UNK-R, as there is no suitable relation in the ontology (for example, a new relation could become one *"done"*).

2.4. Forming a Subset from the Data Corpus for Labeling

To comply with external resource constraints, it is necessary to form a representative subset of objects from the data corpus (hereinafter referred to as a subsample) for pseudolabeling. For each text object, we generated an embedding using the model from [1] and then performed clustering and calculated the geometric medians of the clusters. It was assumed that to preserve the semantic representativeness of the subsample, it is necessary to select objects located within a given radius from the geometric median of each cluster in the space of vector representations of objects in the data corpus.

In the first step, to reduce computational complexity, the embedding dimension was reduced from 768 (the size of the output hidden layer of the model) to 128 using principal component analysis (PCA):

$$X' = \mathrm{PCA}(X), \quad X \in \mathbb{R}^{N\times D},$$

where X is the original embedding matrix of N objects of dimension D, and X' is the matrix after dimensionality reduction.

The HDBSCAN clustering algorithm [26] was then applied to the reduced-dimensional embeddings:

$$L = \mathrm{HDBSCAN}(X', m),$$

where $L = \{-1, 0, 1, \ldots, K-1\}$ is the vector of cluster labels (where -1 means noise), and m is the hyperparameter of the minimum cluster size by the number of objects.

For each detected cluster $C_k \subseteq X'$, where $k = 0, \ldots, K-1$, the geometric median was calculated using the Waitzfeld algorithm [27]:

$$\mu_k = \underset{\mu}{\operatorname{argmin}} \sum_{x \in C_k} \|x - \mu\|_2, \quad \mu_k \in \mathbb{R}^D,$$

which acts as the coordinate of the most typical point of the cluster k.

For each cluster C_k, we selected the radius r_k for all the objects $x_i \in C_k$, satisfying the condition

$$\|x - \mu\|_2 \le r_k,$$

then the total number of selected objects across all clusters does not exceed the limit of 90000 (the limit was set in accordance with external constraints).

Radii r_k can be found either by using an iterative binary search or by choosing a common quantile of the distance from the medians for all clusters.

The final subsample has the form

$$X_{\text{distil}} = \bigcup_{k=0}^{K-1} \{x_i \in C_k : \|x_i - \mu_k\|_2 \le r_k\}.$$

Thus, the procedure can be summarized. First, the embeddings of the source texts were compressed (PCA) then clustered (HDBSCAN) and characterized using the geometric medians of the clusters. Objects for pseudolabeling were selected based on their distance to the median, which is an approach that allows for the efficient formation of a subsample from the original data corpus with minimal loss of representativeness.

3. ARCHITECTURE AND TRAINING OF THE NER/RE MODEL

The presence of 92 categories in the label dictionary indicates the need to reduce computational complexity and compensate for the imbalance of labels in the sample. It was proposed to implement a parallel labeling process using three labels: CLA, REL, TYPE. The NER/RE model consists of three neural networks that receive as inputs the output vector of each token from the last hidden layer of the BERT encoder model (base encoder, Fig. 3). The networks were trained in parallel on the same input token sequence. Two architecture options were considered: fully connected layers and layers with an attention mechanism. For each variant, we tested 2, 4, and 8 layers of neural networks, halving the size of the hidden representation at each layer so long as it remained greater than or equal to the number of outputs of the corresponding label set for each network.

3.1. Loss Function for the NER/RE Problem with Regularization

The representation of tokens and sequences is implemented as follows. Let $X = (x_1, \ldots, x_n)$ be the matrix of the input sequence of tokens. The base encoder produces hidden states $H = f_{\text{enc}}(X) \in \mathbb{R}^{n\times d}$. Each additional neural network maps the hidden state of the output vector of the inner layer (logits of tokens) according to its own separate dictionary of labels:

$$\hat{y}^{(a)} = f_a(h_i) \in \mathbb{R}^{|C_a|}, \quad a \in \{\text{type, class, rel}\}.$$

Additionally, the model architecture allows for the embedding of the entire sequence to be obtained by averaging the states of all tokens, taking into account the attention mask of the base encoder (pooling), with the exception of paddings [28]. Encoding of the entire sequence will be required at the stage of processing UNK sets to form synsets of the ontology extension:

$$h_{\text{seq}} = \frac{\sum_{i=1}^{n} m_i h_i}{\max\left(\sum_{i=1}^{n} m_i, \varepsilon\right)}, \quad \varepsilon = 10^{-9}, \quad m_i \in \{0,1\}.$$

This technique is adopted in BERT models focusing on vector representations of the entire input sequence at once, as it produces more stable and expressive embeddings than encoding only a single CLS token (especially for semantic search and clustering tasks).

Loss function L is the sum of cross-entropy losses across all three networks. Additionally, a special regularization function for logical consistency of predictions has been introduced.

$$\mathcal{L} = \sum_a \mathcal{L}_{\text{head}}^{(a)} + \lambda \mathcal{L}_{\text{reg}}, \quad a \in \{\text{type, class, rel}\},$$

where, for each network, the loss function is calculated using the standard formula

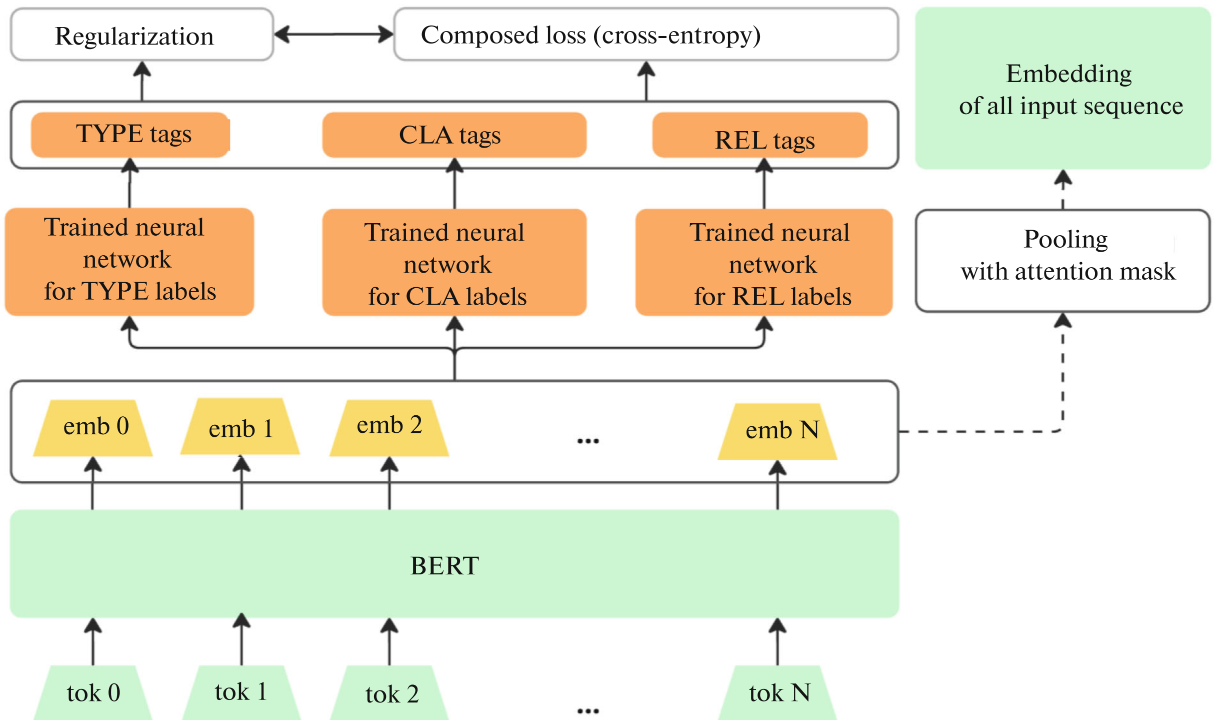

Fig. 3. Architecture of the NER/RE model.

$$\mathcal{L}_{\text{head}}^{(a)} = -\frac{1}{N}\sum_{i=1}^{N} t_i^{(a)\top} \log \operatorname{softmax}\left(y_i^{(a)}\right),$$

$$\left[\operatorname{softmax}\left(y_i^{(a)}\right)\right]_k = \frac{\exp\left(y_{i,k}^{(a)}\right)}{\sum_{j=1}^{K_a}\exp\left(y_{i,j}^{(a)}\right)}.$$

Regulator $\mathcal{L}_{\text{reg}}$ increases the value $\mathcal{L}$ in case the model's predictions fail the consistency check and thus penalizes the model for logical errors in the output label sequences.

Let $\hat{T}_i = \operatorname{argmax} \operatorname{softmax}\left(\hat{y}_i^{(a)}\right)$ be predicted token type i, and $w_i = \max \operatorname{softmax}\left(\hat{y}_i^{(a)}\right)$ evaluates the degree of its prediction (prediction confidence). We introduce predictive labels for classes and relations: C_i and R_i, respectively, as well as the Iverson brackets $[P] \in \{0,1\}$, which are equal to 1 if the prediction is true. a_i denotes the class and b_i denotes the relationship:

$$a_i = [C_i \neq \mathcal{O}], \quad b_i = [R_i \neq \mathcal{O}],$$

where $\mathcal{O}$ is the designation of the O-token (background token assigned the outside class) from the BIO markup. Then the violation indicator for token i equals v_i:

$$\begin{aligned} v_i &= \left[\hat{T}_i = \text{CLA}\right]([C_i = \mathcal{O}] \vee [R_i \neq \mathcal{O}]) \\ &+ \left[\hat{T}_i = \text{REL}\right]([C_i \neq \mathcal{O}] \vee [R_i = \mathcal{O}]) \\ &+ \left[\hat{T}_i = \mathcal{O}\right]([C_i \neq \mathcal{O}] \vee [R_i \neq \mathcal{O}]) \\ &+ \left[\hat{T}_i = \text{UNK}\right](1 - [C_i \neq \mathcal{O}] \oplus [R_i \neq \mathcal{O}]), \end{aligned}$$

where $\vee$ is the logical or and $\oplus$ is the exclusive or (XOR). We can write down $a \oplus b = a + b - 2ab$.

After substitution we get

$$\begin{aligned} v_i &= \left[\hat{T}_i = \text{CLA}\right](1 - a_i(1 - b_i)) \\ &+ \left[\hat{T}_i = \text{REL}\right](1 - b_i(1 - a_i)) \\ &+ \left[\hat{T}_i = \mathcal{O}\right](a_i + b_i - a_i b_i) \\ &+ \left[\hat{T}_i = \text{UNK}\right](1 - a_i - b_i + 2a_i b_i). \end{aligned}$$

Because the events (markup fact) are mutually exclusive, for each token, only one bracket will be active, namely, v_i, which takes the value 1 or 0. The final regularizer with confidence weights on TYPE can be written as

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N} w_i v_i.$$

Let us denote the set of networks as $\mathcal{A} = \{\text{type, class, rel}\}$, and cross-entropy as $CE(t, p) = -t^{\top} \log p$ (for one-hot vector t). Then the loss will look like this:

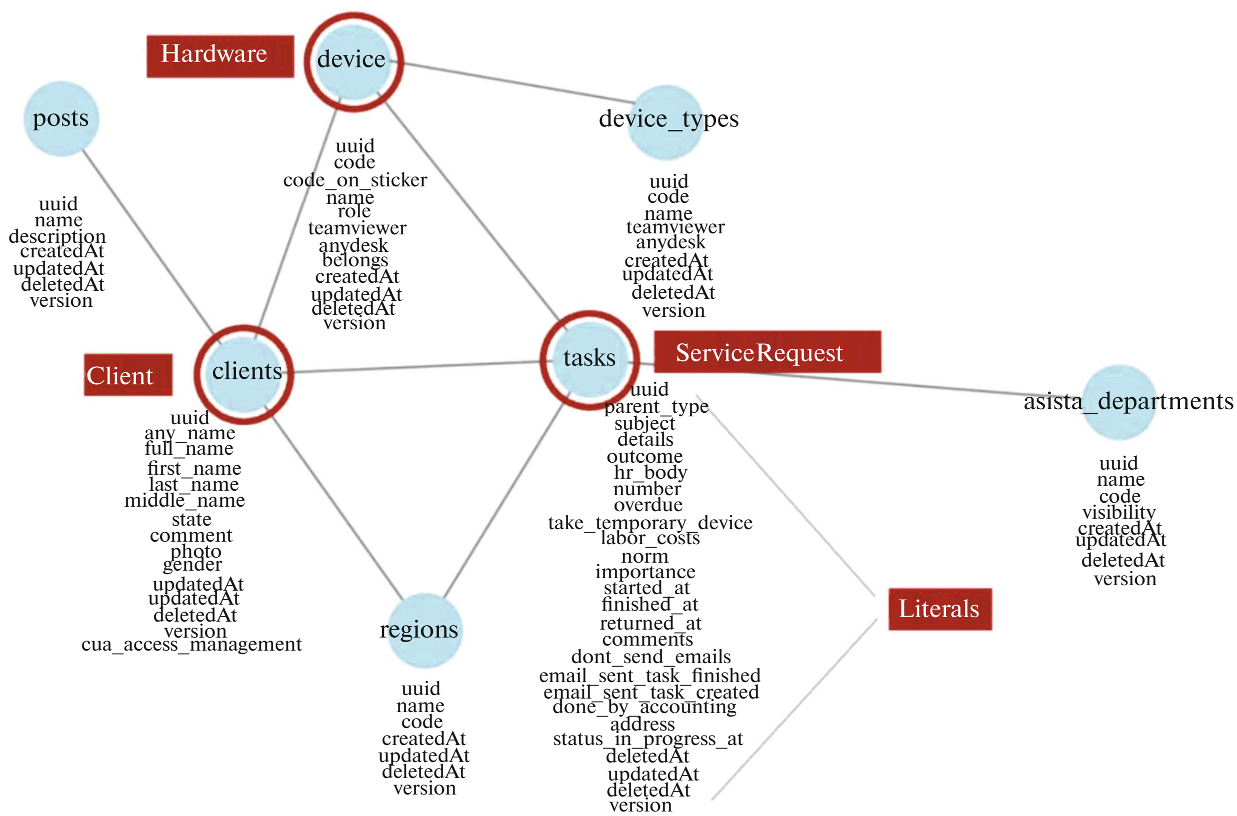

Fig. 4. Mapping scheme of tables from the database to ontology classes.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{a\in\mathcal{A}} CE\left(t_i^{(a)}, \mathrm{softmax}\left(y_i^{(a)}\right)\right) + \lambda w_i v_i\right].$$

This notation makes logical constraints part of the learning goal. Test training cycles were conducted both with the regularizer ($\lambda > 0$), and without it ($\lambda = 0$).

3.2. Model Evaluation

To evaluate the trained NER/RE model, we chose precision (the precision metric) as the optimization metric, which is the proportion of correctly labeled tokens among all tokens to which the model assigned a nonzero (non-“O”) class or relation label. In the context of knowledge graph enrichment, this allows the minimization of the number of false positives in the RDF graph. Each erroneously added entity or relationship breaks semantic consistency and requires manual cleaning, and missed facts can be further extracted in subsequent iterations of the pipeline. This determines the priority of the precision metric with a moderate decrease in recall, i.e., the proportion of correctly labeled tokens of a given class among all tokens of a given class in the sample (hereinafter referred to as the recall metric).

4. CREATION AND ENRICHMENT OF THE KNOWLEDGE GRAPH

To incorporate the extracted facts (in the form of subject-predicate-object triplets) into the knowledge graph, we developed a pipeline that transforms the raw data and BIO markup results into RDF format. In the first step, the main tables were selected from the relational database: *Tasks* (applications), *Companies* (client companies), and *Devices* (equipment). These tables were exported and an instance of the corresponding ontology class was created for each record. So, for example, for a row in a table *Tasks* an instance of the class *ServiceRequest*. Ticket fields (subject, problem description, solution comment, effort, timestamps, etc.) are converted to literals or graph relationships. The mapping of database objects to ontology classes ensures the initial filling of the knowledge graph with structured information from the database, as shown schematically in Fig. 4. At this stage, a subset of 10000 applications was extracted, which were then transformed into an initial graph for subsequent enrichment with the NER/RE model.

In the second step, knowledge extraction from the text was performed using the previously trained NER/RE model (Sec. 4). For each application, text fields (subject, description, and comments) were extracted and combined into one text fragment. The

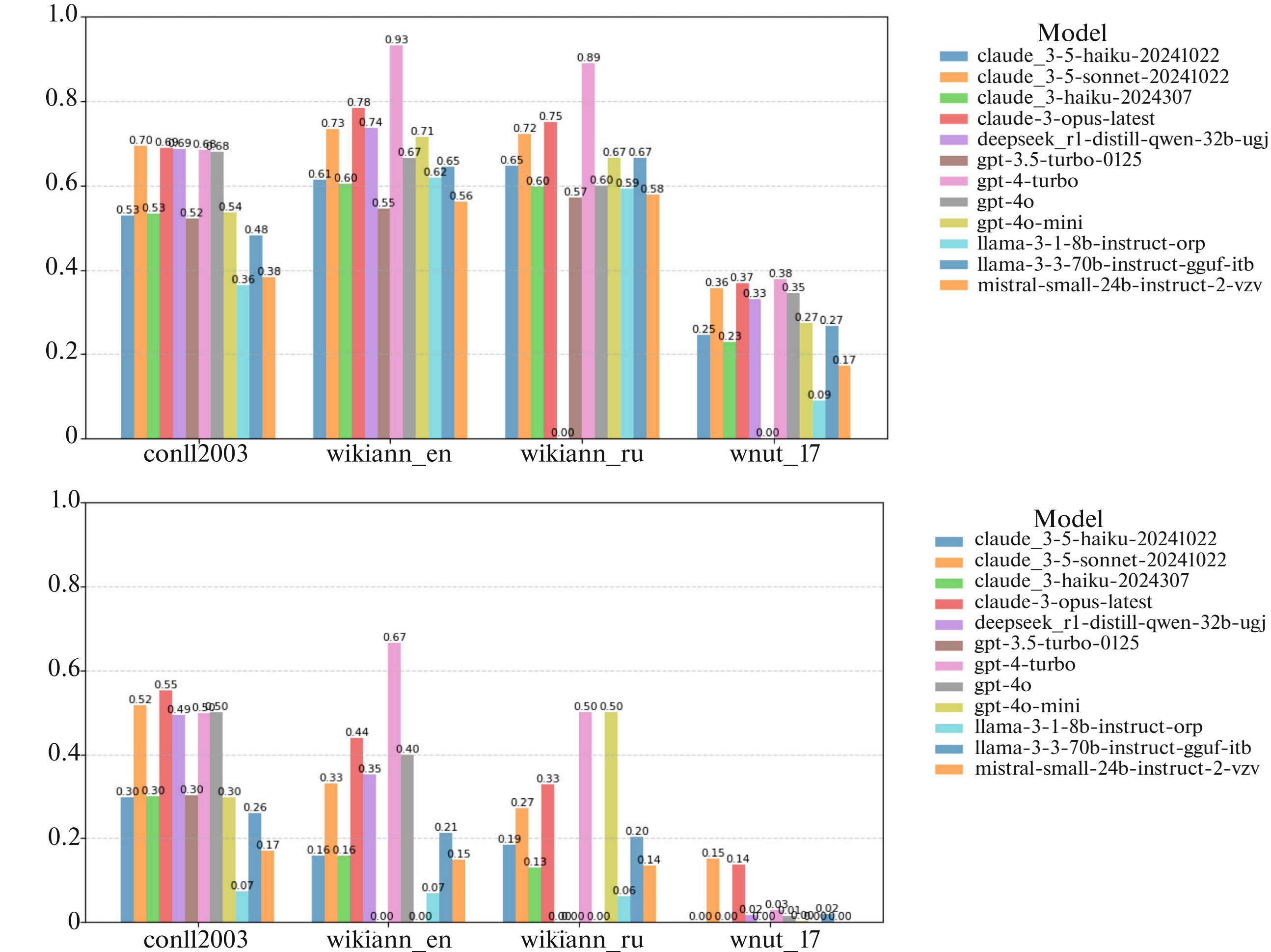

Fig. 5. Comparison chart of Token F1 and Seqeval F1 metrics.

NER/RE model processed a text fragment and returned a sequence of labels for the tokens.

Sequences of tokens of length n with BIO-tags of the same class $\left\{B^c, I_1^c \ldots I_n^c\right\}$ were combined into a single named entity, an instance of the corresponding class or ontology relationship. Each such object was assigned a unique uniform resource identifier (URI, for example http://itog.it/ExtractedEntity/UUID).

Then, based on the predicted relationships, a template was checked: if an entity—relationship—entity sequence was found (for example, "the server provides the application") and the prediction corresponded to an ontological property that is valid between the classes of these entities, then an RDF triple was formed linking the two extracted entities by this property.

5. RESULTS

5.1. Testing and Selecting a Model for BIO Pseudolabeling

The models were compared on four benchmarks (CoNLL 2003, WikiAnn (en), WikiAnn (ru), and WNUT 17) using two types of F1 metrics. Figure 5 shows the distribution diagrams of metrics grouped by datasets. The models are marked in different colors. The top chart shows the Token F1 metric values, and the bottom chart shows the Seqeval F1 values for each model.

The GPT-4-turbo model outperforms its competitors in most cases on the BIO tagging task. Interestingly, this model was not the most modern at the time of the experiments, but it showed the best results. Figure 6 shows a plot of the Seqeval F1 metric averaged across all trials. The worst results were obtained for the WNUT 17 dataset, presumably due to the specificity of the labels: it is more difficult for the model to perform labeling if the labels themselves are represented by large sequences of characters or contain independent meaning.

Thus, the following conclusions can be drawn from the average Seqeval F1 score across all datasets: large models lead in the average score, and the GPT-4-turbo model is among the best in the average score, ranking first in the two WikiAnn datasets. In the English news corpus CoNLL 2003, the first place is

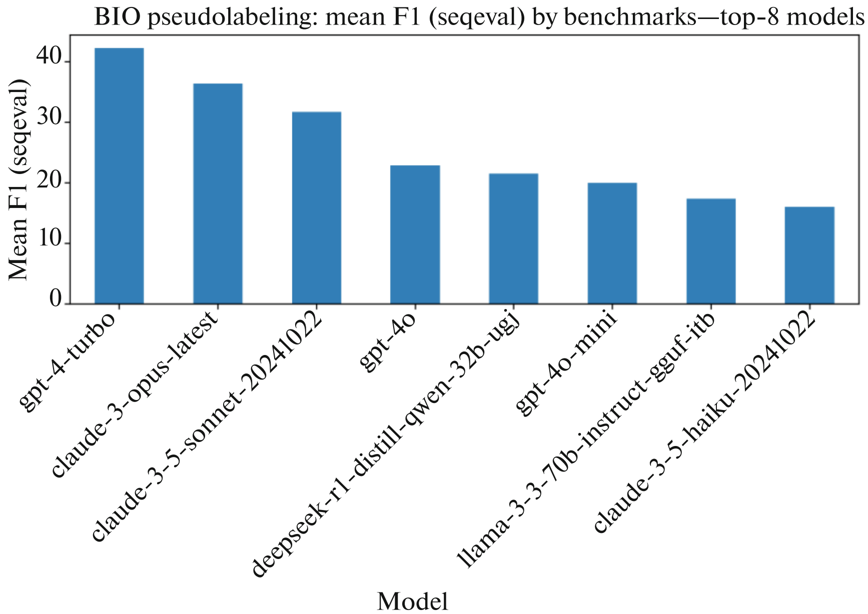

Fig. 6. Comparison of models by average Seqeval F1.

taken by the Claude-3-opus model, while WNUT 17 remains difficult for all contenders from the list of candidate models for BIO-labeling (see Appendix 3).

The experimental results showed that: the best practical choice is the GPT-4-turbo model; it performs stably in terms of multilingual benchmarks, demonstrates high precision metrics, and it yields a better/comparable Seqeval F1 metric on multilingual corpora.

5.2. Formation of a Sample Subset and Pseudolabeling

Although the GPT-4-turbo model demonstrated the best results for the benchmarks, the GPT-4o model was selected for the final dataset labeling. This decision is due to the following factors: (1) significant cost savings with comparable quality for Russian-language data; (2) better performance on multilingual corpora that match the dataset profile; and (3) optimal precision/recall ratio (see Appendix 2).

A rough calculation yielded the number of objects that could be marked within the external constraints: 90000 or about 13% of the original data corpus. Filtration was carried out using the method specified in paragraph 2.4. Figure 7 shows a two-dimensional projection of the clustered embeddings of objects in the original data corpus (left) and the filtered subset (right).

After the GPT-4o model was applied, three subsets of tokens were extracted, labeled as types, classes, and relations. The result was a labeled data corpus with a volume of 81505 objects containing 3037348 tokens, of which 773324 were labeled (a word with any label apart from "O" is considered to be labeled) (see Appendix 4).

5.3. Fine-Tuning the Base Model

The XLM-RoBERTa-large model (24 transformer layers, ~550 million parameters), which was pretrained on the CommonCrawl corpus for 100 languages [29], was selected as the base model. To account for domain-specific characteristics and improve the quality of the model on domain data, we performed additional domain-adaptive fine-tuning. The training corpus for the MLM task (total volume > 650 k samples) was compiled from the original dataset, which consisted of texts from an internal customer support ticket database and documentation.

During training, 15% of the tokens in each sequence were masked, which is the recommended hyperparameter for the masked language modeling task. The training was conducted over 33 epochs. As can be seen from Table 1, loss and perplexity metrics decreased significantly after fine-tuning, indicating an improvement in the model's ability to generate domain-specific vector representations of natural language texts. The fine-tuning process was performed on a single NVIDIA A100 GPU accelerator for 250 h.

5.4. Training the NER/RE Model

At the stage of hyperparameter selection, it was established that the proposed in Section 4 regularization has a negative impact on the metrics (see Appendix 5). Architectures with attention mechanisms showed the best results. Table 2 shows comparative

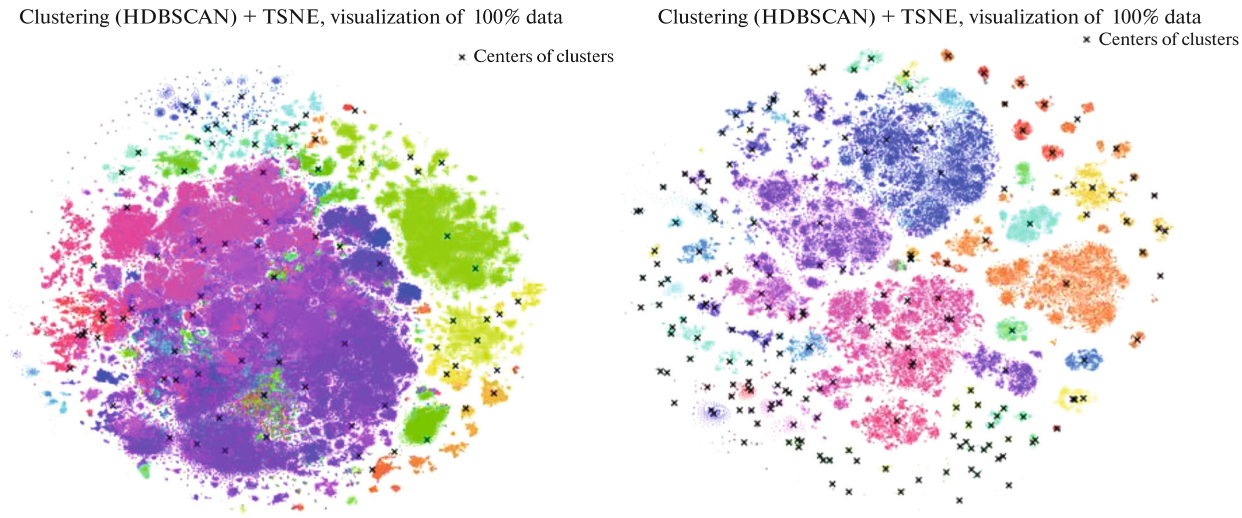

Fig. 7. 2D projection of embeddings before and after filtering.

metrics for the best test training cycles. The head column indicates two types of networks, namely, with an attention mechanism (att, attention) and without it (ff, ffed forward), the "reg" column indicates the regularization coefficient, and "layers" indicates the number of layers in the neural network. The complete table for all test cycles of the NER/RE model training is presented in Appendix 5.

5.5. Model Evaluation

Assessment in Table 2 is performed with respect to the entire incoming sequence of tokens, taking into account the background class "O;" here, the Token F1 metric is also used. This mode demonstrates a high precision metric (≈0.91) with moderate recall values (≈0.54), consistent with the chosen strategy: the model does not strive to predict all potentially informative tokens, but when assigning a CLASS or REL label, it demonstrates high reliability.

After the training was complete, we evaluated the model using the "no O" protocol, i.e., excluding the contribution of background tokens. In this regime, intra-class accuracy and F1 metrics remain high, while the precision metric systematically drops as label support decreases: the expected "long tail" effect. Figure 8 shows the impact of the frequency of labels in the dataset with respect to the metrics. Table 3 shows the most complex labels for the recall and precision metrics, indicating the top errors. The reasons for the errors can be attributed to the semantic closeness of objects and the token-level nature of BIO annotations.

5.6. NER Enrichment of Knowledge Graphs

The overall quantitative and qualitative effects of applying the proposed pipeline to the corporate ITSM base are presented. These results demonstrate not only an increase in the volume of knowledge but also that

Table 1. Comparison of metrics before and after fine-tuning the XLM-RoBERTa-large model on the validation (val.) and test (test.) datasets

	Loss (val.)	Perplexity (val.)	Loss (test.)	Perplexity (test.)
Base model	1.52	4.55	1.53	4.60
Fine-tuned model	0.52	1.68	0.52	1.68

Table 2. Comparative metrics of models

			f1			Accuracy			Precision			Recall		
head	reg	layers	class	rel	avg	class	rel	avg	class	rel	avg	class	rel	avg
ff	0	2	0.56	0.72	0.64	0.45	0.58	0.51	0.85	0.95	0.90	0.45	0.58	0.51
att	0	4	**0.57**	**0.76**	**0.67**	**0.46**	**0.61**	**0.54**	**0.85**	**0.96**	**0.91**	**0.46**	**0.61**	**0.54**

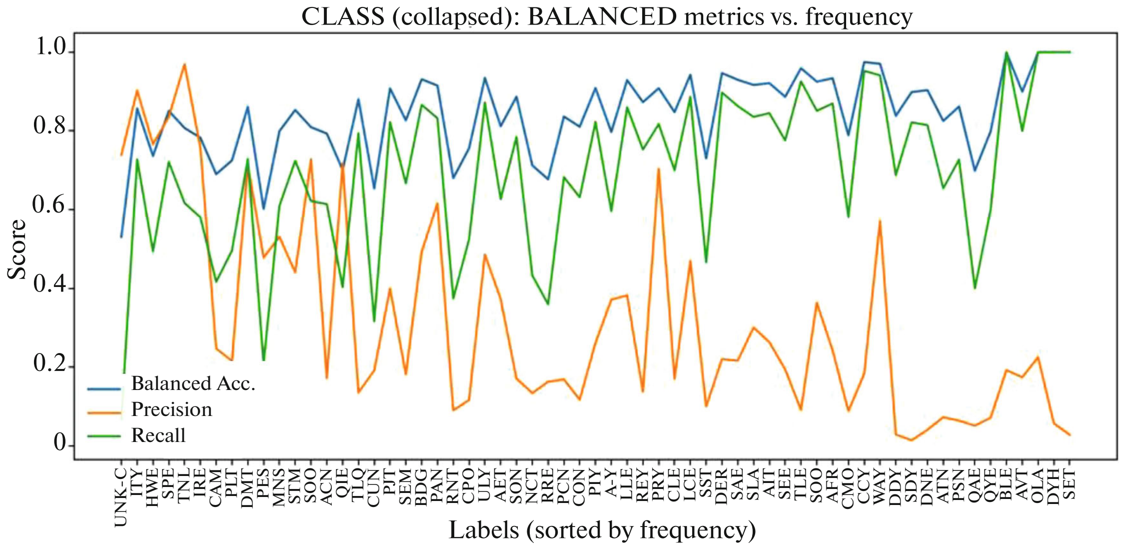

Fig. 8. Graph of metric changes depending on tag frequency.

new entities and relationships fit correctly into the structure of the knowledge graph and remain logically consistent with the constraints of the ontology.

Table 4 shows that at the first stage of processing, each object was transformed into about 12 triplets, where each triplet can be interpreted as a fact obtained from the data.

After the application of the NER model, more than 50 thousand entities were extracted and more than 160 thousand relationships were formed; the automatic inference of new facts multiplies them 2.5-fold. As a result, the total knowledge extracted from 10000 support tickets approaches one million triplets, and the expansion factor increases from 2.65 to 3.52. The proportion of inferred facts reaches approximately 71.6% of the entire graph, indicating an intensive expansion, and automatically inferred facts make up the majority of the graph. At the same time, the integrity of the graph structure was preserved: as before, all nodes are connected into a single component (thanks to the class nodes from the ontology).

Thus, each original row of the database is transformed into hundreds of interconnected facts; this significantly increases the expressiveness of the data and opens up the possibility of complex queries in SPARQL that are not available either at the SQL language level or in the knowledge graph in its original state.

Table 5 shows that the increase in nodes and edges is accompanied by the expected "sparseness" of the network, but the appearance of nonzero clustering (0.08) shows that new entities form related semantic clusters. This means that the new nodes form local groups. For example, mentions of names of software could be linked through the common class *Software Package* or, through one device, through the class *Hardware*.

The resulting semantic network exhibits the properties of a scale-free structure: the distribution of node degrees decreases monotonically, and the distribution includes nodes with relatively high degrees—these are ontology nodes and the most common entities, such as, for example, typical services; the majority of nodes have low degrees (tens of times smaller). This character indicates the correctness of the integration of new data: they did not turn the graph into a chaotic structure where everything is connected to everything else but rather fit into the existing semantic structure, forming semantic clusters around known concepts.

Figure 9 compares the same entity corresponding to the original data object (support ticket) before and after NER enrichment. As can be seen, in the initial state, a limited set of objects are associated with the central node, but after enrichment, nine new connections appeared that were not in the database. They describe time intervals, modes, organizations, responsible persons, etc.; all of this important information was contained implicitly (not explicitly) in the text of the application in natural language.

The model automatically detected instances of classes, after which relationships were established between them in accordance with the axioms that are explicitly contained in the ontology. The cluster and semantic analysis of entities with category labels UNK-C and UNK-R will further allow us to identify new classes and relationships for expanding the ontology.

5.7. Processing UNK Entities and Expanding the Ontology

All entities with the UNK-C (class candidates) and UNK-R (relationship candidates) labels were clus-

Table 3. List of the most difficult labels and the percentage of errors in the “no O” assessment protocol

Labels hardest by recall			Labels hardest by precision		
label	type	confusion (% err)	label	type	confusion (% err)
PES	Class	PCS (26.9%)	CMO	Class	SDY (27.8%)
RRE	Class	STM (22.0%)	TLE	Class	AET (25.0%)
RNT	Class	DDY (13.6%)	**SST**	**Class**	**SDY (40.4%)**
QIE	**Class**	**CCY (38.4%)**	**CPO**	**Class**	**SDY (68.6%)**
CAM	Class	HWE (19.5%)	CON	Class	CUN (17.3%)
UNK-R	Relation	O (14.9%)	**HRS**	**Relation**	**HER (42.1%)**
HSR	Relation	HRS (60.9%)	HCB	Relation	HER (25.0%)
HRS	**Relation**	**HER (42.1%)**	HIE	Relation	HLY (33.3%)
RIS	**Relation**	**UES (22.6%)**	HAT	Relation	HIS (16.7%)
HES	Relation	HRS (20.9%)	UES	Relation	HIE (15.0%)
HIS	Relation	HES (21.3%)	HME	Relation	HES (33.3%)
DEF	**Relation**	**O (37.5%)**	**HVE**	**Relation**	**HRS (40.0%)**

Table 4. Comparison of database and knowledge graphs

Data	Source and volume	Nodes	Explicit triplets	Withdrawn	Total facts	Expansion ratio[1]
SQL	10 000 rows of tasks (+keys inside rows)	≈10 000	–	–	–	–
Basic RDF	Direct SQL → RDF mapping	13 899	115 661	191 262	306 923	2.65
NER- RDF	Base graph + automatically extracted facts	66 194	279 546	705 330	984 876	3.52

[1]The expansion ratio is calculated as the proportion of the total number of facts in the graph to explicit triplets (those that were explicitly imported into the graph), and it is an important indicator of the automatic inference of new facts.

Table 5. Topological changes of the graph after NER enrichment

Metrics	Before NER	After NER	Δ
Number of nodes	13 899	66 194	+376%
Number of ribs	40 094	151 670	+278%
Average degree $\langle k \rangle$	5.77	4.58	–21%
Density	4.1×10^{-4}	6.9×10^{-5}	↓×6
Clustering $\overline{C}$	0.00	0.08	+0.08

tered into synsets suitable for expert validation and inclusion in the ontology.

For each UNK entity, embeddings were constructed in the manner specified in Section 4.

The initial set contained 244 679 elements. In the first step, we removed noise and nonword units and filtered by cosine similarity, setting the threshold as the 10th percentile of the distribution of distances within pairs separately for classes and for relationships. After this procedure, 50 630 class candidates and 141 317 relationship candidates remained. Additional cleaning by token length (at least three characters) reduced the sample to 44 631 and 112 695 objects, respectively. Using principal component analysis (PCA), the embeddings were then reduced to 128 dimensions, after which clustering was performed using the k-medium (minibatch) k-means) with hyperparameter $k = 12$ for each set of objects. For each cluster, the geometric median was calculated using the Weitzfeld algorithm; a compact neighborhood of fixed size (no more than the 50 closest objects) was formed around this median, the original objects of which formed a synset for expert analysis. Figure 10 shows the effect of geometric median filtering on clusters of

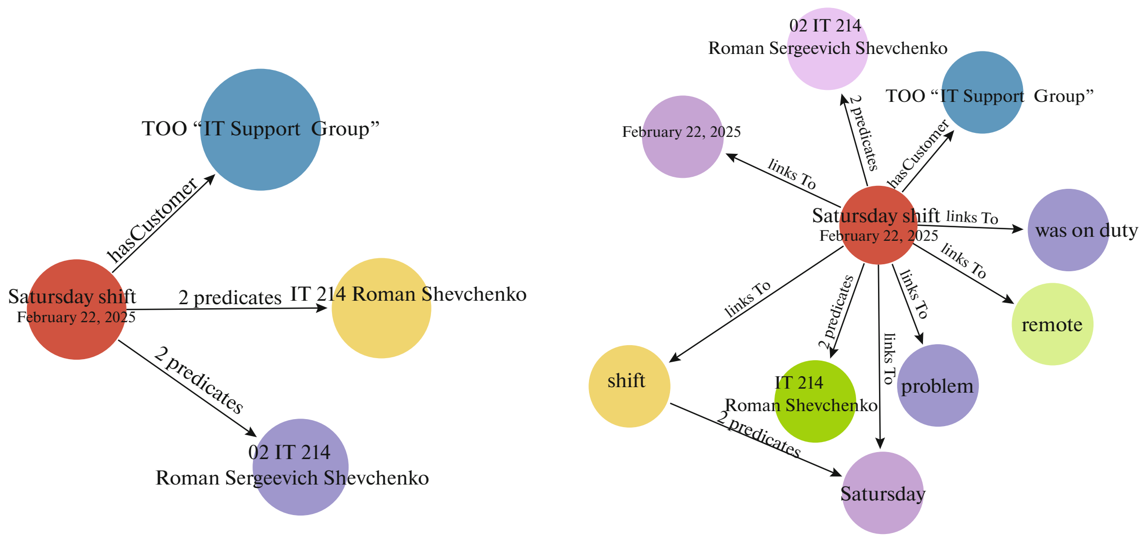

Fig. 9. Fragment of the knowledge graph before and after NER enrichment.

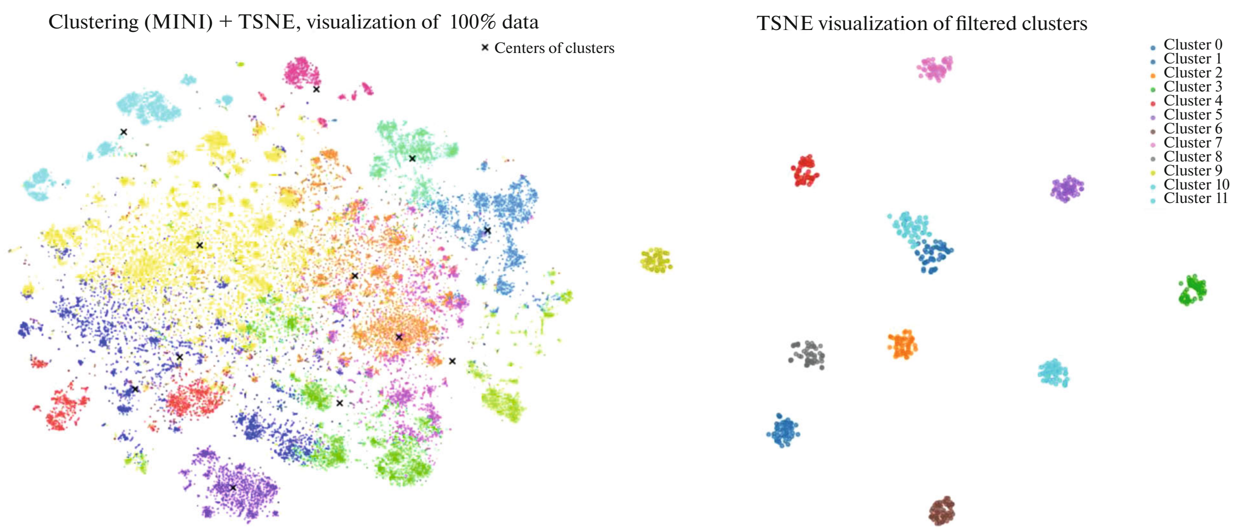

Fig. 10. 2D projection of UNK embeddings before and after filtering.

objects labeled as UNK-CLA (unknown class representatives extracted from input texts) and presents a 2D projection before and after filtering. As a result, we obtained dense and homogeneous local semantic sets (Appendix 6).

A qualitative analysis of synsets conducted by an expert demonstrated the semantic coherence of the groups and allowed the proposal of ontological specifications. Thus, the synset in which verb forms such as "issued," "changed," "saved," "tested," and "restored" predominate was interpreted as a class of modification processes: *rdfs:label—ModificationProcess; rdfs:comment—"the type of process responsible for modifying configuration items, settings, and parameters."* Properties such as "responsibility" and "role" are in this case inherited from the ontology and are necessary to complement RDF triples.

6. PRACTICAL SIGNIFICANCE AND USE SCENARIO

Automating knowledge graph enrichment significantly reduces the labor intensity of maintaining knowledge relevance in ITSM. Unlike the manual addition of ontology by experts, the proposed method can be implemented as a background process with minimal intervention. Integrating NER/RE results with ontological constraints ensures a high accuracy of

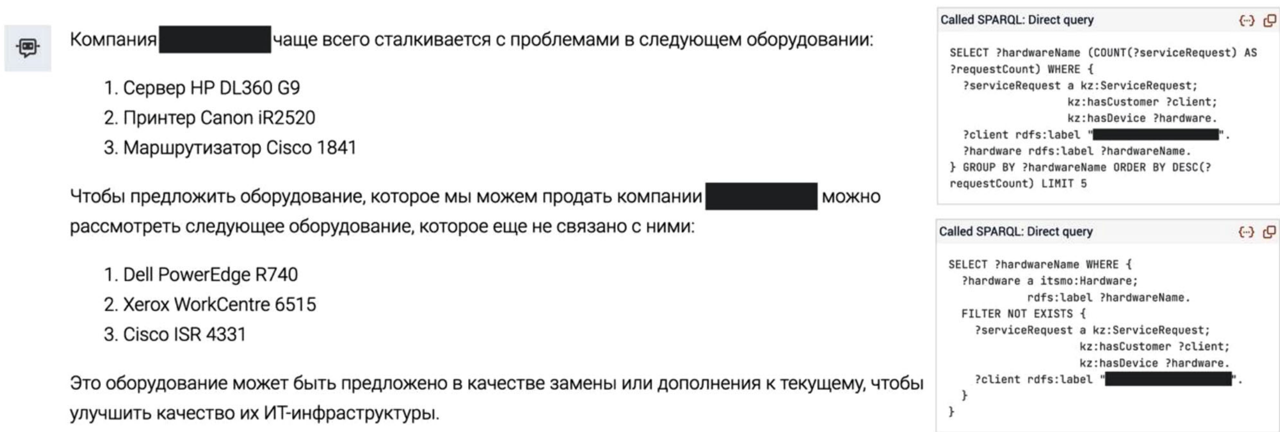

Fig. 11. Interface for processing queries to a knowledge graph in natural language.

knowledge extraction from texts: the model implements a conservative extraction strategy and adds to the graph only those facts that it is confident in and that match the schema. Thanks to this property, semantic consistency of data is achieved: the system does not violate the rules and restrictions defined by the ontology. It is significant that, even after an almost fivefold increase, the graph retains a strict structure and connectivity.

The automated graph satisfies specific queries that are not achievable in the relational data model. Figure 11 shows the GraphDB Chat interface: an LLM connected to enriched graph answers the question *“What breaks most often at company XXXXXX and what can you offer them?”* Next, the graph facts are used to derive the client’s most problematic devices and generate a proposal relevant to the client’s behavior, stored as a linked set of facts in the knowledge graph.

Stages of response formation:

(1) The SPARQL subquery retrieves aggregated breakdown statistics (COUNT for entities of the Hardware class associated with ServiceRequest and ClientCompany).

(2) The second query searches for equipment of the same type that is not available at the customer (FILTER DOES NOT EXIST { ?new hw itsmo:hasCustomer :XXX }).

(3) LLM interprets the results by ranking them by frequency.

In a tabular SQL schema, this synthesis is complicated: facts concerning the state of the equipment are stored in rows of different tables, and the concept of missing but compatible equipment is not explicitly represented. The graph model, together with a specialized language model, allows this condition to be specified using a SPARQL construct. Thus, we demonstrate that our pipeline improves query expressivity and opens the way to recommendation scenarios without manual preparation of data marts.

CONCLUSION

A reproducible graph—model—graph cycle for the automatic enrichment of the corporate knowledge graph and semiautomatic expansion of the ontology in the ITSM domain has been developed and experimentally validated. The key components of the proposed approach are:

- a unified ontological scheme that combines the DOLCE ontology with the ITSMO domain ontology;
- significant BIO-corpus (3 million tokens, 92 tags);
- original architecture of the NER/RE model;
- UNK-mark mechanism.

The ontology was expanded with 12 new classes and 12 new relationships. The inclusion was carried out with a consistency check according to the hierarchies and constraints of the ontology.

The proposed cycle—from markup to ontologically formulated constructs—is based on the priority of accuracy, as is noted in Section 4: We deliberately choose a conservative model with a high precision metric because for semiautomated integration, the reliability of a single solution is more important than exhaustive coverage. In practice, this strategy reduces the proportion of false positive candidates in UNK sets, reduces the workload for the expert, and speeds up the UNK—synset—specification—inclusion steps.

Taken together, this demonstrates that UNK tags, enhanced by contextual embeddings and compact cluster sampling, serve as an effective mechanism for identifying previously absent concepts and properties. The method provides a reproducible, controllable, and reactive ontology extension in a semiautomated manner.

The effectiveness of the approach is confirmed by quantitative metrics: a 4.76-fold increase in the graph size (from 13899 to 66194 nodes) and an increase in the expansion coefficient from 2.65 to 3.52 while maintaining logical consistency. UNK tags for semiautomatic detection of new classes and relationships can significantly reduce the time required to update ontologies.

A set of methods for creating synthetic data sets using an ontology dictionary as a label space is presented, the possibilities of explicit and implicit logical regularizations in the process of training an encoder model are demonstrated, and the research results are tested in working scenarios.

APPENDIX 1

A prompt used to test models' BIO-markup abilities. All class labels from the class dictionary and the text for marking are supplied to sorted_labels and text, respectively.

```
Below is a sentence. You have to annotate each token using BIO format for named entity recognition.
The possible entity classes are: {sorted_labels}
Use the 'O' tag if the token does not belong to any entity.
Sentence: {text}
Return the annotations as valid JSON with a single key 'ner', whose value is a list of the same length as the token list.
Each element in that list should be a BIO tag (eg, 'B-PER', 'I-PER', 'O').
For example: [\"B-PER\", \"O\", \"B-LOC\", ...]
```

APPENDIX 2

Prompt used by LLM pseudomarkup. The prompt fragment for multiclass NER markup consists of two main parts: there is a part with examples given in the required JSON format, the examples of which contain all types of labels, and the second part is the prompt itself, which contains the workload in the form of text to be marked up and a list of labels with a description of the semantics of each label (the prompt does not specify the annotations of each class).

```
few_shot_examples = (
    Example 1:
    Input: Windows doesn't work, I can't connect to the database.
    Output (JSON):
    {
      "ner": {
        "Not": "O",
        "works": "O",
        Windows: "B-SPE",
        ",": "O",
        "Not": "O",
        "it works out": "O",
        "connect": "B-RIS",
        "To": "O",\n'
        "base": "B-UNK-C" // unknown class (eg data-base)
      }
    }

system_prompt = (
    "You're an expert NER and Relation Extraction model specialized in the IT domain and ITIL. Annotate each token using BIO
tagging for both Named Entity Recognition (NER) and relation extraction."

[ENTITY & RELATION LABELS]
    {entity_labels_str}
[END LABELS]

    Instructions:
    - Assign 'B-' for the first token of an entity or relation.
    - Assign 'I-' for subsequent tokens within the same en-tity or relation.
    - Use 'O' for tokens not belonging to any entity or re-lation
    - Prioritize tagging IT and ITIL domain-specific terminology.
    - Try to label as many words as possible. If unsure, use 'UNK'.
    Output format (JSON):
    {
      \"ner\": {
        \"token1\": \"BIO-tag\",
        token2\": \"BIO-tag\",
        ...
      }
    Examples:
    {few_shot_examples}
    Proceeded with provided text following the same schema.
)
```

APPENDIX 3

Results of an experiment that tests different models on BIO benchmarks.

Benchmark	Model	F1		Precision		Recall	
		seq.	bin	seq.	bin	seq.	bin
conll2003	claude-3-5-haiku-20241022	29.9%	52.8%	27.2%	52.2%	33.1%	53.4%
	claude-3-5-sonnet-20241022	51.6%	**69.5%**	47.4%	**66.7%**	56.6%	72.5%
	claude-3-haiku-20240307	30.1%	53.5%	28.0%	51.7%	32.4%	55.3%
	claude-3-opus-latest	**55.2%**	69.0%	**53.6%**	65.5%	**56.9%**	73.0%
	deepseek-r1-distill-qwen-32b-ugj	49.3%	68.6%	45.1%	61.2%	54.3%	**78.2%**
	gpt-3.5-turbo-0125	30.3%	52.3%	30.4%	52.2%	30.2%	52.3%
	gpt-4-turbo	49.7%	68.4%	48.0%	62.3%	51.6%	75.9%
	gpt-4o	50.2%	67.9%	47.5%	63.1%	53.1%	73.6%
	gpt-4o-mini	29.9%	53.6%	28.0%	49.2%	32.0%	58.9%
	llama-3-1-8b-instruct-orp	7.3%	36.4%	5.3%	25.2%	11.7%	65.8%
	llama-3-3-70b-instruct-gguf-itb	26.0%	48.2%	23.8%	44.5%	28.7%	52.5%
	mistral-small-24b-instruct-2-vzv	17.1%	38.3%	15.2%	35.8%	19.6%	41.1%
wikiann_en	claude-3-5-haiku-20241022	15.8%	61.3%	14.3%	75.0%	17.8%	51.8%
	claude-3-5-sonnet-20241022	33.1%	73.3%	29.7%	85.1%	37.5%	64.4%
	claude-3-haiku-20240307	15.8%	60.4%	14.1%	74.8%	17.9%	50.7%
	claude-3-opus-latest	44.0%	78.3%	40.8%	86.7%	47.7%	71.3%
	deepseek-r1-distill-qwen-32b-ugj	35.1%	73.6%	31.9%	84.9%	39.1%	64.9%
	gpt-3.5-turbo-0125	0.0%	54.5%	0.0%	75.0%	0.0%	42.9%
	gpt-4-turbo	**66.7%**	**93.3%**	**66.7%**	**87.5%**	**66.7%**	**100.0%**
	gpt-4o	40.0%	66.7%	50.0%	80.0%	33.3%	57.1%
	gpt-4o-mini	0.0%	71.4%	0.0%	71.4%	0.0%	71.4%
	llama-3-1-8b-instruct-orp	6.9%	61.8%	5.2%	57.5%	10.2%	66.9%
	llama-3-3-70b-instruct-gguf-itb	21.3%	64.5%	18.4%	74.1%	25.3%	57.1%
	mistral-small-24b-instruct-2-vzv	15.0%	56.1%	13.5%	71.8%	16.8%	46.1%
wikiann_ru	claude-3-5-haiku-20241022	18.5%	64.8%	15.9%	74.8%	22.1%	57.1%
	claude-3-5-sonnet-20241022	27.1%	72.2%	23.1%	83.8%	32.6%	63.5%
	claude-3-haiku-20240307	13.1%	59.8%	11.2%	69.5%	15.8%	52.4%
	claude-3-opus-latest	32.9%	75.0%	29.5%	84.5%	37.4%	67.4%
	deepseek-r1-distill-qwen-32b-ugj	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	gpt-3.5-turbo-0125	0.0%	57.1%	0.0%	66.7%	0.0%	50.0%
	gpt-4-turbo	**50.0%**	**88.9%**	**50.0%**	**100.0%**	**50.0%**	**80.0%**
	gpt-4o	0.0%	60.0%	0.0%	60.0%	0.0%	60.0%
	gpt-4o-mini	50.0%	66.7%	50.0%	60.0%	50.0%	75.0%
	llama-3-1-8b-instruct-orp	6.1%	59.3%	4.4%	52.4%	9.6%	68.4%
	llama-3-3-70b-instruct-gguf-itb	20.4%	66.6%	17.3%	73.6%	24.9%	60.9%
	mistral-small-24b-instruct-2-vzv	13.5%	57.9%	12.3%	71.7%	15.0%	48.6%
wnut	claude-3-5-haiku-20241022	0.0%	24.5%	0.0%	17.9%	0.0%	39.0%
	claude-3-5-sonnet-20241022	**15.2%**	35.6%	10.6%	26.5%	**27.1%**	54.2%
	claude-3-haiku-20240307	0.0%	23.0%	0.0%	16.1%	0.0%	40.3%
	claude-3-opus-latest	13.6%	36.7%	**9.6%**	26.2%	23.3%	61.5%
	deepseek-r1-distill-qwen-32b-ugj	1.7%	33.0%	1.1%	22.2%	3.3%	**64.4%**
	gpt-3.5-turbo-0125	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	gpt-4-turbo	2.8%	**37.7%**	2.1%	**26.8%**	4.5%	63.2%
	gpt-4o	1.4%	34.5%	1.0%	24.1%	2.3%	60.7%
	gpt-4o-mini	0.2%	27.4%	0.1%	20.2%	0.3%	42.8%
	llama-3-1-8b-instruct-orp	0.1%	9.0%	0.1%	4.9%	0.6%	55.3%
	llama-3-3-70b-instruct-gguf-itb	2.0%	26.8%	1.3%	19.1%	3.7%	44.6%
	mistral-small-24b-instruct-2-vzv	0.0%	17.3%	0.0%	12.4%	0.0%	28.7%

APPENDIX 4

Distribution of labels in the synthesized dataset

[distribution cat_rel_none]:
O: 2232028
CLA: 667908›
REL: 105496
UNK: 31916

[Distribution of class_tags]:

O: 2369495	SOQ: 8429	PCS: 2136	CLE: 815	DNE: 303
UNK-C: 125902	ACN: 6258	ULY: 1992	SAE: 736	ATN: 258
ITY: 100113	QIE: 5575	SON: 1954	SLA: 698	PSN: 229
HWE: 81238	TLQ: 4470	CON: 1808	CCY: 651	SDY: 213
SPE: 80533	CUN: 3938	RRE: 1769	TLE: 632	BLE: 151
TNL: 62252	PJT: 3242	PIY: 1472	SOO: 564	AVT: 146
IRE: 49713	PAN: 2772	PRY: 1124	DDY: 515	SET: 115
CAM: 18107	SEN: 2545	LLE: 1076	AFR: 513	OLA: 102
PLT: 16653	BDG: 2539	DER: 974	AIT: 511	QYE: 94
PES: 15758	CPO: 2384	SST: 969	SEE: 481	DYH: 82
DMT: 14996	NCT: 2249	LCE: 935	CMO: 437	
MNS: 13858	RNT: 2179	AY: 927	WAY: 409	
STM: 1295	AET: 2159	REY: 873	QAE: 370	

[Distribution of rel_tags]:

O: 2931877	HOE: 1536	HGN: 679	HCB: 269	_A: 100
RIS: 39290	HRS: 1494	HPS: 671	HIE: 227	HIY: 67
UNK-R: 30466	HSD: 1255	HVE: 511	HER: 184	HRY: 19
HSR: 15628	HNE: 1157	HME: 469	NO: 174	HNH: 18
HIS: 2782	HNS: 1131	HGS: 440	HOY: 168	
HES: 2361	HFD: 863	DEF: 313	HCN: 143	
UES: 1933	HAT: 720	HEN: 279	HLY: 12	

APPENDIX 5

Synsets and their transformation into an ontology object

Example of a class and its definition:

['issued',
'changed',
'did',
'Prescribed',
'Sent',
'changed',
'saved',
'Changed',
'Prescribed',
'Changed',
'Protested',
'Changed',
'Brought out',
'Cancel',
'saved',
'dropped',
'Changed',
'Created',
'changed',
'Requested',
'introduced',
'gave',
'Sent',
'Send',
'Readdressed',
'will be applied',
'Throwed down',
'changed',
'Added',
'I found out',
'dropped',
'Sent',
'Changed',
'I found out',
'Freed',
'changes',
'Checked',
'restored',
'Requested',
'added',
'Sent',
'saved',
'I'll rename',
'Changed',
'Cancel',
'Chosen',
'Asks',
'changed',
'Cancel',
'Enabled']

An example of a relationship and its definition:

['Dmitry.',
'absent.',
'absent.',
'UEFI,',
'absent.',
'absent.',
'absent.',
'195.',
'Mail.',
'NdqkyRI.',
'evernot.',
'absent.',
'Ugpx.',
'check.',
'Int.',
'delivered.',
'laptop.',
'Skype.',
'okay,',
'Planned.',
'issue.',
'those.',
'Bitrix.',
'absent.',
'week.',
'absent.',
'teamweaver:.',
'absent.',
'modem.',
'reconnection',
'Daria.',
'hSPjds.',
'keys.',
'outlook.',
'folder.',
'Planned.',
'Dmitry.',
'Jimail.',
'Sultan.',
'evernot.',
'absent.',
'absent.',
'forwarding:',
'email.',
'year.',
'Zhanatov.',
Astana,
'lift.',
'works.',
'ev.'],

rdfs:label: ModificationProcess
rdfs:comment: A type of process that handles the modification or change of a configuration item, setting, or parameter.
Potential Properties: hasResponsible, hasAccountable, hasInformed, etc. (already defined in ITSMO for processes)
owl:DatatypeProperty: hasNote
rdfs:comment: Holds additional notes or short remarks about the resource’s current status or environment.
<owl:DatatypeProperty rdf:about="http://ontology.it/itsmo/v1#hasNote">
<rdfs:domain rdf:resource="http://ontology.it/itsmo/v1#RunnableResource"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:label>hasNote</rdfs:label>
<rdfs:comment>Additional short remark or note about a re-source.</rdfs:comment>
</owl:DatatypeProperty>

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Khalov, A. and Ataeva, O., Automating ontology mapping in IT service management: A DOLCE and ITSMO integration, *Data Sci. J.*, 2025, vol. 24, p. 23. https://doi.org/10.5334/dsj-2025-023
2. Borgo, S., Ferrario, R., Gangemi, A., Guarino, N., Masolo, C., Porello, D., Sanfilippo, E.M., and Vieu, L., DOLCE: A descriptive ontology for linguistic and cognitive engineering1, *Appl. Ontol.*, 2023, vol. 17, no. 1, pp. 45–69. https://doi.org/10.3233/ao-210259
3. IT Service Management Ontology (ITSMO), Canonical resolver; catalog entry in LOV IT Service Management Ontology (ITSMO). https://w3id.org/itsmo.ontology.it.lov.linkeddata.es. Cited August 8, 2025.
4. Gruber, T.R., A translation approach to portable ontology specifications, *Knowledge Acquisition*, 1993, vol. 5, no. 2, pp. 199–220. https://doi.org/10.1006/knac.1993.1008
5. Gruber, T.R., Toward principles for the design of ontologies used for knowledge sharing?, *Int. J. Hum.-Comput. Stud.*, 1995, vol. 43, nos. 5–6, pp. 907–928. https://doi.org/10.1006/ijhc.1995.1081
6. Smith, B., Ontology (science), *Formal Ontology in Information Systems*, Eschenbach, C. and Grüninger, M., Eds., IOS Press, 2008, pp. 21–35.
7. Studer, R., Benjamins, V.R., and Fensel, D., Knowledge engineering: Principles and methods, *Data Knowl. Eng.*, 1998, vol. 25, nos. 1–2, pp. 161–197. https://doi.org/10.1016/s0169-023x(97)00056-6
8. El Yamami, A., Mansouri, Kh., Qbadou, M., and Illoussamen, E., An ontological representation of ITIL framework service level management process, *Smart Data and Computational Intelligence*, Khoukhi, F., Bahaj, M., and Ezziyyani, M., Eds., Lecture Notes in Networks and Systems, vol. 66, Cham: Springer, 2019, pp. 88–94. https://doi.org/10.1007/978-3-030-11914-0_9

9. Barrasa, J. and Webber, J., *Building Knowledge Graphs: A Practitioner's Guide*, O'Reilly, 2023.

10. Hogan, A., Blomqvist, E., Cochez, M., et al., *Knowledge Graphs*, Morgan Claypool, 2021.

11. Valiente, M.-C., Vicente-Chicote, C., and Rodríguez, D., An ontology-based and model-driven approach for designing IT service management systems, *International Journal of Service Science, Management, Engineering, and Technology*, 2011, vol. 2, no. 2, pp. 65–81. https://doi.org/10.4018/jssmet.2011040104

12. Miwa, M. and Bansal, M., End-to-end relation extraction using LSTMs on sequences and tree structures, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, 2016, Erk, K. and Smith, N.A., Eds., Association for Computational Linguistics, 2016, vol. 1, pp. 1105–1116. https://doi.org/10.18653/v1/p16-1105

13. Xu, J., Zhang, Z., Friedman, T., Liang, Yi., and Van den Broeck, G., A semantic loss function for deep learning with symbolic knowledge, *Proceedings of the 35th International Conference on Machine Learning*, Dy, J. and Krause, A., Eds., Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 5502–5511. https://proceedings.mlr.press/v80/xu18h.html.

14. Sun, K., Zhang, R., Mensah, S., Mao, Yo., and Liu, X., Learning implicit and explicit multi-task interactions for information extraction, *ACM Trans. Inf. Syst.*, 2023, vol. 41, no. 2, p. 27. https://doi.org/10.1145/3533020

15. Giunchiglia, E. and Lukasiewicz, T., Coherent hierarchical multi-label classification networks, *Advances in Neural Information Processing Systems*, 2020, vol. 33. https://proceedings.neurips.cc/paper/2020/file/6dd4e10e3296fa63738371ec0d5df818-Paper.pdf.

16. Yu, J., Bohnet, B., and Poesio, M., Named entity recognition as dependency parsing, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., Eds., Association for Computational Linguistics, 2020, pp. 6470–6476. https://doi.org/10.18653/v1/2020.acl-main.577

17. Lu, Ya., Liu, Q., Dai, D., Xiao, X., Lin, H., Han, X., Sun, L., and Wu, H., Unified structure generation for universal information extraction, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, 2022, Muresan, S., Nakov, P., and Villavicencio, A., Eds., Association for Computational Linguistics, 2022, vol. 1, pp. 5755–5772. https://doi.org/10.18653/v1/2022.acl-long.395

18. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N.A., Don't stop pretraining: Adapt language models to domains and tasks, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., Eds., Association for Computational Linguistics, 2020, pp. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

19. Brown, T.B. et al., Language models are few-shot learners, *Advances in Neural Information Processing Systems*, 2020, vol. 33. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

20. Alizadeh, M., Kubli, M., Samei, Z., Dehghani, Sh., Zahedivafa, M., Bermeo, J.D., Korobeynikova, M., and Gilardi, F., Open-source LLMs for text annotation: A practical guide for model setting and fine-tuning, *J. Comput. Soc. Sci.*, 2025, vol. 8, no. 1, p. 17. https://doi.org/10.1007/s42001-024-00345-9

21. Eiras, F., Petrov, A., Vidgen, B., Schroeder De Witt, Ch., Pizzati, F., Elkins, K., Mukhopadhyay, S., Bibi, A., Csaba, B., Steibel, F., Barez, F., Smith, G., Guadagni, G., Chun, J., Cabot, J., Imperial, J.M., Nolazco-Flores, J.A., Landay, L., Jackson, M.T., Rottger, P., Torr, P., Darrell, T., Lee, Yo.S., and Foerster, J.N., Position: Near to mid-term risks and opportunities of open-source generative AI, *Proceedings of the 41st International Conference on Machine Learning*, Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., Eds., Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 12348–12370. https://proceedings.mlr.press/v235/eiras24b.html.

22. Tjong, E.F. and De Meulder, F., Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, Edmonton, Canada, 2003, Stroudsburg, PA: Association for Computational Linguistics, 2003, vol. 4, pp. 142–147. https://doi.org/10.3115/1119176.1119195

23. Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H., Cross-lingual name tagging and linking for 282 languages, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, 2017, Barzilay, R. and Kan, M.-Y., Eds., Association for Computational Linguistics, 2017, vol. 1, pp. 1946–1958. https://doi.org/10.18653/v1/p17-1178

24. Derczynski, L., Nichols, E., van Erp, M., and Limsopatham, N., Results of the WNUT2017 shared task on novel and emerging entity recognition, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, 2017, Derczynski, L., Xu, W., Ritter, A., and Baldwin, T., Eds., Association for Computational Linguistics, 2017, pp. 140–147. https://doi.org/10.18653/v1/w17-4418

25. Brown, T.B. et al., Language models are few-shot learners, *Advances in Neural Information Processing Systems*, 2020, vol. 33. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

26. Campello, R.J.G.B., Moulavi, D., Zimek, A., and Sander, J., Hierarchical density estimates for data clustering, visualization, and outlier detection, *ACM Trans. Knowl. Discovery Data*, 2015, vol. 10, no. 1, p. 5. https://doi.org/10.1145/2733381

27. Vardi, Ye. and Zhang, C.-H., A modified Weiszfeld algorithm for the Fermat–Weber location problem, *Math. Program.*, 2001, vol. 90, no. 3, pp. 559–566. https://doi.org/10.1007/pl00011435

28. Reimers, N. and Gurevych, I., Sentence-BERT: Sentence embeddings using Siamese BERT-networks, *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019, Inui, K., Jiang, J., Ng, V., and Wan, X., Eds., Association for Computational Linguistics, 2019, pp. 3982–3992. https://doi.org/10.18653/v1/d19-1410

29. Hugging Face. XLM-RoBERTa (large): specs (24 layers, ~550M params), 2020–2024. https://huggingface.co/transformers/v3.4.0/pretrained_models.html.
30. Côté, M.-A., Kádár, Á., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., El Asri, L., Adada, M., Tay, W., and Trischler, A., TextWorld: A learning environment for text-based games, *Computer Games. CGW 2018*, Cazenave, T., Saffidine, A., and Sturtevant, N., Eds., Communications in Computer and Information Science, vol. 1017, Cham: Springer, 2018, pp. 41–75. https://doi.org/10.1007/978-3-030-24337-1_3
31. Russell, S. and Norvig, P., Automated planning, *Artificial Intelligence: A Modern Approach*, Pearson, 2020, 4th ed.
32. Schmidhuber, J., Gödel machines: Self-referential universal problem solvers making provably optimal self-improvements, *Artificial General Intelligence*, Goertzel, B. and Pennachin, C., Eds., Cognitive Technologies, Berlin: Springer, 2007, pp. 199–226. https://doi.org/10.1007/978-3-540-68677-4_7
33. Yin, X., Wang, X., Pan, L., Lin, L., Wan, X., and Wang, W.Ya., Gödel agent: A self-referential agent framework for recursively self-improvement, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, Vienna, 2025, Che, W., Nabende, J., Shutova, E., and Pilehvar, M.T., Eds., Association for Computational Linguistics, 2025, pp. 27890–27913. https://doi.org/10.18653/v1/2025.acl-long.1354
34. Ataeva, O.M. and Serebryakov, V.A., Ontology of the digital semantic library LibMeta, *Informatika i Ee Primeneniya*, 2018, vol. 12, no. 1, pp. 2–10. https://doi.org/10.14357/19922264180101