Language Model-Based Algorithm for Constructing Knowledge Graphs from Patent Data

Nikita Gavrilov[1(✉)], Vladimir Korkhov[1], Evgenii Pen[2], and Alexey Tokarev[3]

[1] Saint Petersburg State University, Saint Petersburg, Russia
st102441@student.spbu.ru, v.korkhov@spbu.ru
[2] BioGeoHab, Saint Petersburg, Russia
[3] EDRID, Saint Petersburg, Russia

Abstract. This paper presents a method for constructing a knowledge graph based on patent data, which facilitates the identification of hidden relationships between patents and the organization of information for subsequent analysis. The method involves extracting key textual fields from patent documents and vectorizing them using state-of-the-art transformer models, and building a graph where the nodes represent individual documents, and the edges reflect their semantic proximity. A clustering algorithm is employed to group the patents, ensuring high internal coherence within clusters and reducing the original graph to a compact representation. The resulting clusters are summarized using language models, enabling automatic extraction of significant terms for cluster descriptions. Experimental research conducted on a large corpus of patent data demonstrates the efficacy of the proposed approach, which is confirmed by the relevant partitioning quality metrics. The proposed method improves the interpretation of patent information, facilitating the identification of implicit relationships and structural patterns, which is of great importance for analyzing scientific achievements and managing intellectual property.

Keywords: Knowledge Graph · Patent Data · Text Vectorization · Clustering

1 Introduction

Knowledge graphs are data structures that represent relationships between various entities and are utilized to address a wide range of tasks, such as information retrieval, recommendation systems and natural language processing. In contrast to traditional knowledge graphs, which contain specific entities as nodes and express relationships through triplets of the form ⟨entity, relationship, entity⟩ [1], in this approach, the nodes of the graph contain textual data, and the relationships between them are identified using vector representations of these texts.

The relevance of this research is underscored by the growing volume of patent data, which is an important source of information for analyzing scientific and technological achievements. Patents contain both structured and unstructured

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
O. Gervasi et al. (Eds.): ICCSA 2025 Workshops, LNCS 15894, pp. 219–230, 2026.
https://doi.org/10.1007/978-3-031-97648-3_15

information about new technologies, inventions, and their legal protection, rendering them a valuable resource for analyzing technological trends and innovations. However, the substantial volume of this data and its complex structure complicate manual processing and analysis, necessitating the development of automated knowledge extraction methods.

The primary challenge in constructing knowledge graphs from patent data lies in the complexity of the text, which often contains specialized terminology and legal phrasing. Extracting entities from such texts necessitates meticulously annotated data. To tackle this challenge, it is essential to clearly define the application domain of a specific solution, which involves standardizing the problem formulation to ensure its wide applicability. Additionally, manual text annotation is a labor-intensive process that requires substantial time and resources, limiting the creation of a sufficient volume of annotated data for training models.

To efficiently process such data, it is necessary to develop algorithms that account for these features, particularly for the Russian language, where existing approaches designed for English are not always applicable [2].

The contribution of this paper is a flexible and lightweight method for constructing knowledge graphs from patent data. Unlike conventional approaches that rely on named entity recognition (NER) and relation extraction (RE) to extract specific entities and their relationships, our method models entire documents as graph nodes and infers their connections based on semantic similarity using transformer-based language models. This enables scalable graph construction without the need for pre-trained NER/RE pipelines, making the method language-agnostic, easy to implement, and particularly effective in low-resource languages or domains where high-quality annotation tools are unavailable.

The paper is structured as follows: Sect. 2 reviews related work on language models. Section 3 formulates the problem and outlines key challenges. Section 4 presents the proposed graph construction and clustering method. Section 5 analyzes the experimental results, and Sect. 6 concludes the paper and suggests directions for future research.

2 Related Work

Constructing knowledge graphs based on patent data is based on the task of finding relevant patents, which is a key area of application for information retrieval methods.

To solve the task of finding relevant patents, it is reasonable to use models with transformer architecture. Multilingual models, such as mBERT, XLM-R [3] and mDeBERTa [4], are successfully applied to process the Russian language, but with the development of monolingual models, their role has become supplementary. Among Russian-language models, RuBERT [5] and RuLeanALBERT [6], trained on corpora such as Wikipedia, web texts, and social media, stand out. Monolingual models have significantly improved the understanding of Russian text. Let us consider several popular models for processing the Russian language:

- The ruBERT model is based on the BERT architecture [7], which uses two key pretraining objectives: the masked language modeling (MLM) task and the next sentence prediction (NSP) task;
- ruRoBERTa is based on the RoBERTa architecture [8] and uses a modified version of the MLM pretraining task. Unlike ruBERT, the ruRoBERTa model was trained on a larger dataset, significantly improving its performance. The model uses Byte-level BPE (BBPE) tokenization and has a vocabulary size of 50,000 tokens;
- The ruELECTRA model [9] uses the ELECTRA architecture and follows the replaced token detection (RTD) task, where the model is trained to predict which tokens were replaced by the MLM-based generator.
- ruT5 is one of the first models for the Russian language based on the T5 architecture [10], which employs an encoder-decoder approach. The model is trained on the task of span corruption, where consecutive spans of text are masked, and the model learns to recover the masked tokens using SentencePiece tokenization.
- FRED-T5 (Full-scale Russian Enhanced Denoisers) is a model based on T5 and UL2 [11], which uses multiple pretraining objectives to create a more versatile approach. Unlike ruT5, FRED-T5 uses the gated GELU activation function instead of ReLU and is trained on a mixture of different types of denoisers.

New multilingual models with Russian language support continue to emerge. EuroBERT [12], for instance, is a multilingual encoder built on the Llama 3 architecture. It has been trained on a large multilingual dataset that includes programming code and mathematical texts and is available in three sizes (210M, 610M, and 2.1B parameters). The training process follows a two-stage approach, consisting of pre-training and annealing, with masked language modeling (MLM) as the core objective. Several architectural refinements have been implemented, including grouped query attention, swish gated linear units, root mean square layer normalization, rotary position embeddings, and the removal of bias terms.

The development of models with encoder architecture improves the accuracy of text processing and analysis, enhancing semantic understanding. This directly impacts the effectiveness of patent search tasks and knowledge graph construction. The same applies to large language models, which significantly improve the capabilities of text data analysis and interpretation.

Large language models, such as Llama 3 [13], fine-tuned on Russian-language text corpora, effectively solve NER tasks and annotation creation. However, in this work, computational power limitations prevent the use of any models, making the development of compact language models for specialized tasks both important and relevant today.

3 Problem Statement

The goal of this work is to propose a method for identifying relationships between patent data. To achieve this goal, the following tasks have been formulated:

- Development and implementation of a graph construction algorithm;
- Application of a clustering algorithm to the graph;
- Interpretation of the obtained results.

The specific technical challenges addressed in this research include:

- Semantic representation: Selecting and implementing appropriate transformer models to convert patent texts into vector representations that accurately capture their technical content, and utilizing specialized vector databases for efficient similarity comparisons.
- Graph construction optimization: Determining the optimal parameters for graph construction, particularly the number of connections (k) between nodes, to balance connectivity and computational efficiency.
- Cluster granularity: Finding the appropriate resolution parameter for the clustering algorithm that produces meaningful patent groupings without creating excessively large or numerous clusters.
- Cluster interpretation: Creating an automated method to generate descriptive labels for patent clusters that accurately represent their technical content.

4 Proposed Solution

Figure 1 illustrates the process of converting the initial set of patent data into a graph through the sequential execution of four stages:

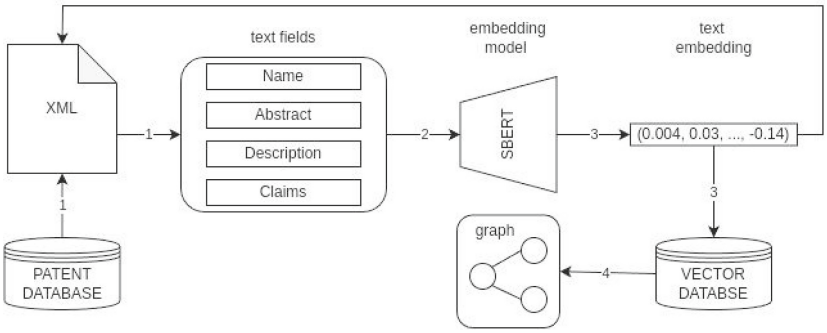


Fig. 1. Graph Construction Algorithm.

1. The main textual fields of each patent are extracted, stored in XML format. For further vectorization, we will use the claims field, as it is considered the most informative and conveys the essence of the patent.

2. Each of the extracted text blocks undergoes vectorization. For this, a transformer-based model that supports Russian either as the main language or as one of the possible languages is used. We chose ruRoBERTa-large because it demonstrates high performance on the Russian SuperGLUE tasks [14].
3. A vector database is employed for efficient storage and retrieval of documents. In our implementation, we use the FAISS library, which supports GPU processing.
4. A graph node is assigned to each patent, and edges between nodes represent the semantic proximity of the documents. Semantically similar patents are determined based on a fixed number of nearest neighbors or a threshold value of cosine distance.

The constructed patent graph serves as the basis for further analysis. To identify structural patterns and group similar documents, the Leiden clustering algorithm [15] is applied, ensuring high internal coherence of the nodes. The algorithm consists of four stages:

1. Local node movement. Each node is initially placed in its own community, after which iterative movement of nodes to neighboring communities takes place. A node moves if such an action increases the objective function (e.g., modularity), which improves the partitioning quality.
2. Partition refinement. At this stage, the obtained partitioning is refined: each community is analyzed for internal coherence. If there are parts within a community that are weakly connected, the community can be split into several more compact and well-connected sub-communities.
3. Graph aggregation. After partition refinement, each community is reduced to a single supernode, and the edge weights between supernodes are determined as the sum of the original edge weights between the corresponding communities. The algorithm then repeats on the aggregated graph until a stable partitioning is reached.

The resulting clusters are summarized using large language models, which automatically extract significant terms and generate meaningful descriptions. Based on these descriptions, a new graph is formed, where each node represents an individual cluster. As a result, the original graph size is reduced.

5 Results Analysis

For the study, a corpus of 329,826 patents, represented in XML format, from the RUPAT database provided by the Federal Institute of Industrial Property (FIPS) was used. The RUPAT database contains full texts of Russian patents, including descriptions of inventions, claims, abstracts, as well as data about applicants and authors. The textual fields of the patent documents were extracted and stored in the PostgreSQL.

An experiment was conducted to determine the optimal number of neighbors, where the parameter k (the number of neighbors for each node) was analyzed for

its impact on the structure of the graph. The experiment involved incrementally increasing k in steps of 2, starting from the minimum value. As k increases, the density of connections rises, which improves coherence, but it can also lead to data redundancy. The objective of the experiment was to ascertain the optimal value of k at which each node would maintain sufficient connectivity with other nodes, ensuring that the graph does not contain isolated subgraphs. This is crucial, as the connectivity of the graph significantly influences the subsequent clustering process. Additionally, the relationship between unique and duplicate connections with respect to k was investigated. The aim was to determine the value of k at which redundancy in the graph begins to increase. As the number of neighbors increases, the number of unique connections should remain stable, while the number of duplicate connections increases, which may indicate rising redundancy in the graph. The results of the experiment are shown in Fig. 2.

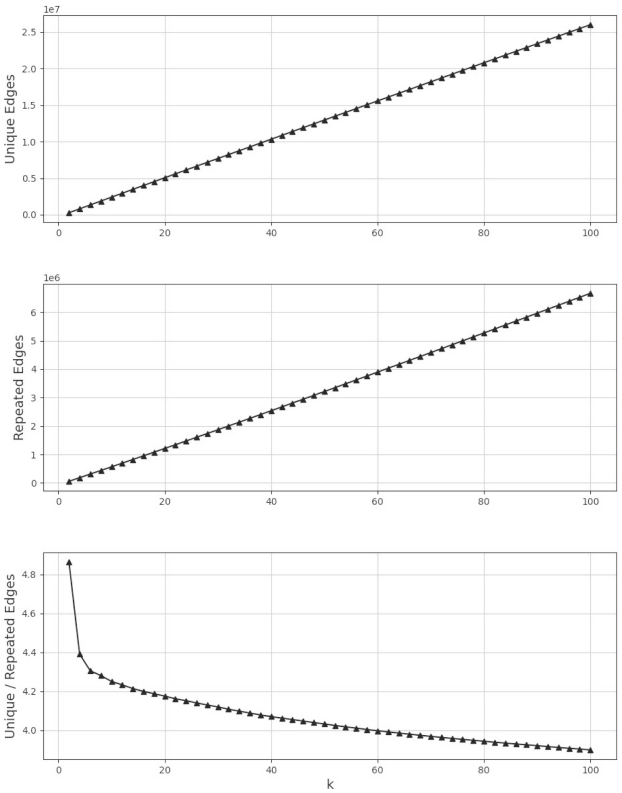


Fig. 2. Dependence a) of unique, b) of duplicate, and c) of the ratio of unique to duplicate connections on the parameter k

As seen in Fig. 2, there is no stabilization in the number of unique connections or an increase in duplicates. A similar pattern remains for the number of

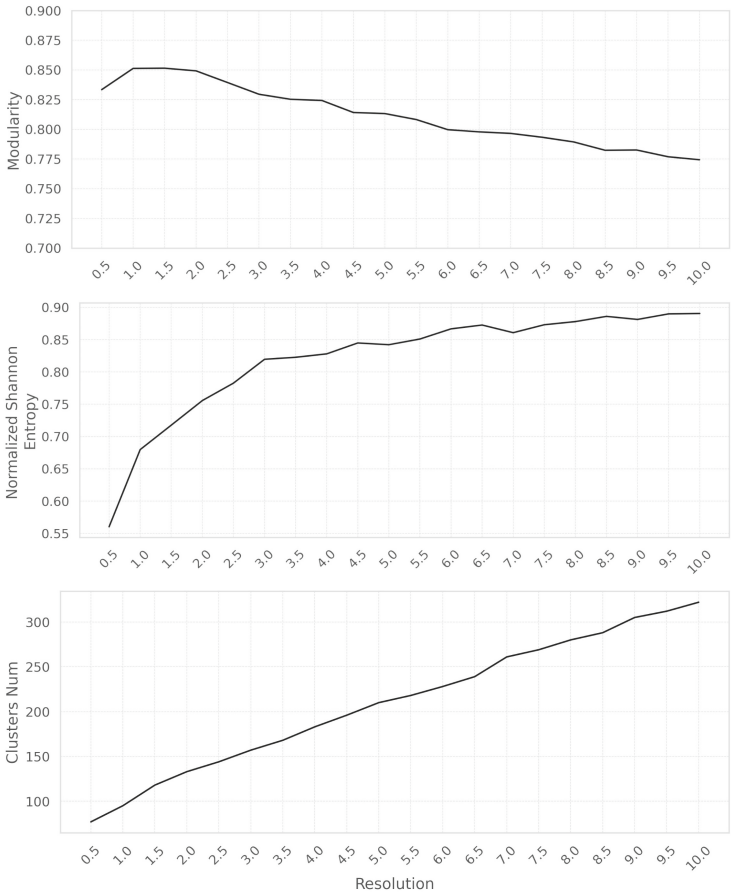


Fig. 3. Dependence a) of modularity, b) of Shannon entropy, and c) of the number of clusters on the resolution parameter

connections at 200, 300, 400, 500, and 600. Further relationships could not be established due to the memory limitation of the graphics card. Therefore, the only option is to select k for which the graph connectivity condition holds. Consequently, the first graph built using the algorithm from the previous section contains 329826 nodes and 2403178 edges, in this implementation of the algorithm, k=10, which guarantees graph connectivity.

The uniformity of the clustering plays an important role since excessively large or sparse clusters hinder further analysis and reduce informativeness. Therefore, in Fig. 3, the results of changes in the modularity metric, calculated using formula (1), normalized Shannon entropy, and the number of clusters as a function of the resolution parameter are presented. The resolution parameter allows controlling the granularity of the clustering.

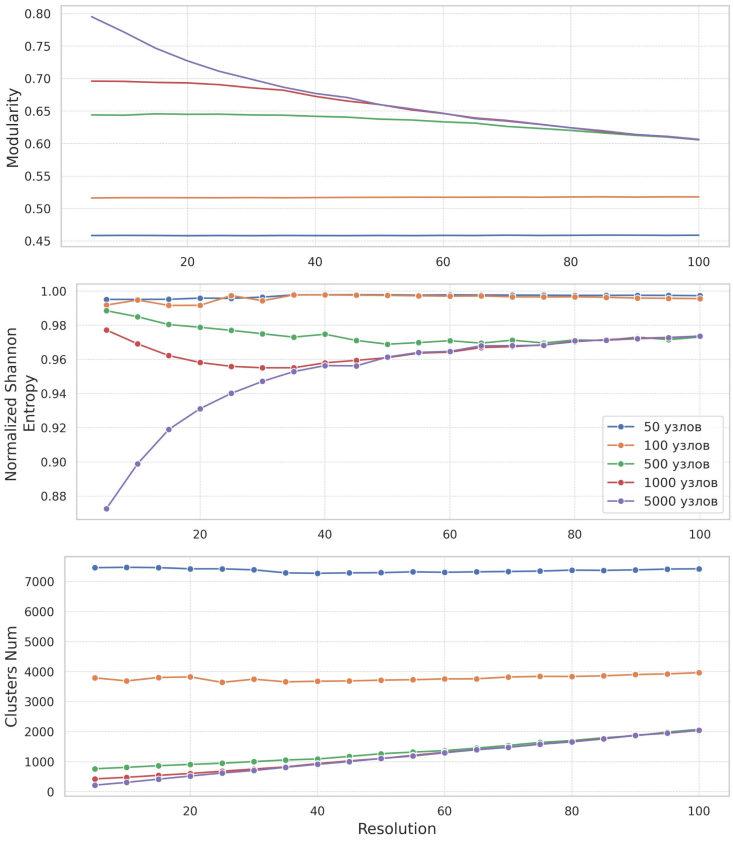


Fig. 4. Dependence a) of modularity, b) of Shannon entropy, and c) of the number of clusters on resolution for different values of max_comm_size

$$Q = \frac{1}{2\,m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2\,m} \right) \delta(c_i, c_j), \tag{1}$$

where:

- m is the total number of edges in the graph;
- A_{ij} is the element of the adjacency matrix, equal to 1 if there is an edge between nodes i and j, and 0 otherwise;
- k_i и k_j are the degrees of nodes i and j (the number of edges incident to the nodes);
- $\delta(c_i, c_j)$is the Kronecker delta function, equal to 1 if nodes i and j are in the same community, and 0 otherwise.

Figure 4 shows the changes in the same metrics, but for a specified limit on the number of nodes in each cluster: the implementation of the algorithm in the

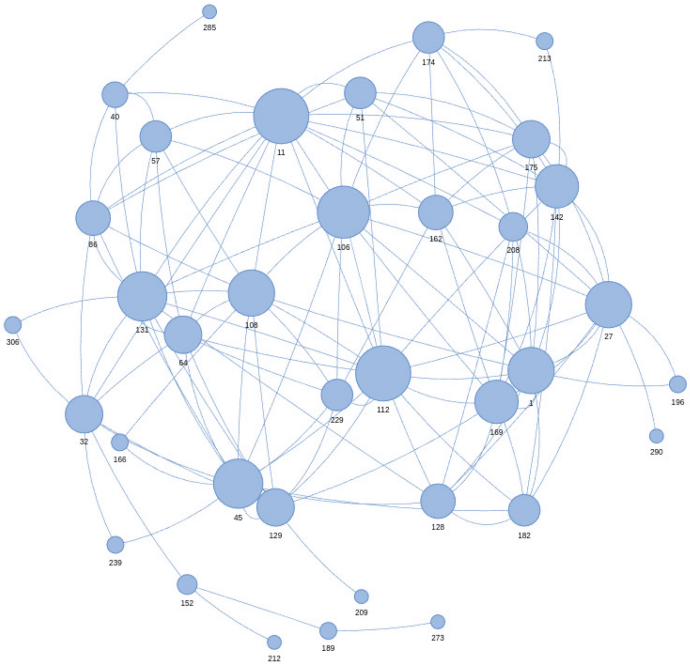


Fig. 5. Subgraph of the final graph

Leiden library [16] allows setting the parameter max_comm_size (Maximum community size), which restricts the cluster size.

Modularity optimization algorithms, such as Leiden, do not always perform well at detecting small clusters if their size is below a certain threshold [17]. Therefore, the modularity metric is not always informative when determining the quality of partitioning.

In Figs. 3 and 4, we observe that as the number of clusters increases, their distribution becomes more uniform, but at the same time, the modularity metric decreases, which, as shown earlier, is not critical. We establish that a normalized Shannon entropy value of 0.9 is acceptable, and based on this, we set the resolution parameter to 10, which ensures a more even distribution of nodes across clusters. As a result, we obtain 322 clusters with a maximum number of nodes in a cluster equal to 4529, a normalized Shannon entropy of 0.89, and a modularity value of 0.77.

Thus, the new graph contains 322 nodes and 7370 edges: two clusters are connected if there is at least one edge in the original graph linking nodes from these clusters. Figure 5 shows the subgraph, where the size of a node represents the number of patents in the cluster. The weight of each edge in the new graph is determined by the number of edges in the original graph that connect nodes from the two respective clusters. The use of a large language model, in our case,

```
{
    "18": {
        "keyword": "Hydraulic control systems",
        "ids": [
            "2700483",
            "2532720",
            "2792767",
            "2477682",
            "2753012"
        ]
    },
    "34": {
        "keyword": "Optical sensors",
        "ids": [
            "2795677",
            "2538293",
            "2504043",
            "2786357",
            "2543693"
        ]
    },
    "40": {
        "keyword": "Chemical compositions",
        "ids": [
            "2537622",
            "2489501",
            "2722305",
            "2616629",
            "2574616"
        ]
    }
}
```


Fig. 6. Example of JSON structure for cluster descriptions

Mistral 7B, fine-tuned on a Russian text corpus [18], allows for the extraction of key terms that describe the clusters.

Due to computational resource limitations, we decided to randomly select 5 texts from each cluster to construct the descriptions. This not only reduces the load on the model but also serves as an additional check on the quality of the clustering. The model we use, with 7 billion parameters, is not always capable of effectively analyzing texts from various domains, especially when they differ significantly in terms of subject matter. In cases where the selected subset contains patents with very different content, the model may incorrectly identify key terms and form a poor description. Below is the structure of the JSON file(see Fig. 6) that stores the key terms and patent identifiers for the selected clusters used to generate their descriptions.

A detailed analysis and interpretation of the obtained results require further investigation. It is expected that using the complete set of texts from each cluster will provide a more accurate characterization of its content; however, implementing this approach requires the development of specialized data processing methods. The application of a large language model may only offer a preliminary representation of a cluster. Additionally, each patent contains metadata, including information about the authors, patent holders, the patent's start date, as well as textual fields such as the title, abstract, and description. This data can be used for a more detailed analysis and interpretation of the cluster structure.

6 Conclusion

This paper presents an algorithm for building a knowledge graph based on patent data, which enables the identification of semantic relationships between patents. The use of text vectorization and graph clustering with the Leiden algorithm helps to uncover hidden structural patterns and simplifies the analysis of large volumes of data. The results suggest that the proposed approach enhances the interpretation of patent data, providing a more compact and informative representation.

Unlike traditional approaches that focus on extracting entity-level triplets, our method constructs graphs at the document level, offering a broader view of technological domains by linking semantically similar patents. This makes the approach especially effective for exploratory analyses and low-resource language settings. However, it lacks the fine-grained specificity of classical knowledge graphs. In future work, we plan to combine our method with entity-level extraction techniques to create multi-resolution representations that capture both general structure and detailed relationships.

References

1. Bruches, E., Pauls, A., Batura, T., Isachenko, V.: Entity recognition and relation extraction from scientific and technical texts in Russian. arXiv preprint arXiv:2011.09817 (2020)
2. Siddharth, L., Blessing, L.T.M., Wood, K.L., Luo, J.: Engineering knowledge graph from patent database. arXiv preprint arXiv:2106.06739 (2021)
3. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2020)
4. He, P., Gao, J., Chen, W.: DeBERTaV3: improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv:2111.09543 (2023)
5. Kuratov, Y., Arkhipov, M.: Adaptation of deep bidirectional multilingual transformers for Russian language. arXiv preprint arXiv:1905.07213 (2019)
6. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2020)

7. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2019)
8. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
9. Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
10. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2023)
11. Tay, Y., et al.: UL2: unifying language learning paradigms. arXiv preprint arXiv:2205.05131 (2023)
12. Boizard, N., et al.: EuroBERT: scaling multilingual encoders for European languages. arXiv preprint arXiv:2503.05500 (2025)
13. Grattafiori, A., Dubey, A., Jauhri, A., et al.: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
14. Zmitrovich, D., et al.: A family of pretrained transformer language models for Russian. arXiv preprint arXiv:2309.10931 (2024)
15. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Sci. Rep. **9**(1), 41695 (2019). https://doi.org/10.1038/s41598-019-41695-z
16. Leidenalg. https://leidenalg.readthedocs.io/en/stable/index.html. Accessed 15 Feb 2025
17. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proc. Nat. Acad. Sci. **104**(1), 36–41 (2007). https://doi.org/10.1073/pnas.0605965104
18. Saiga/Mistral 7B, Russian Mistral-based chatbot. https://huggingface.co/IlyaGusev/saiga_mistral_7b_lora. Accessed 17 Feb 2025