Scientific article
UDC 330.47
DOI: https://doi.org/10.57809/2025.4.2.13.1

INTEGRATING GENERATIVE AI FOR TECHNOLOGICAL TREND ANALYSIS AND PATENT RESEARCH AUTOMATION

Vasiliy Pochetniy ✉

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ pochetnyj.va@edu.spbstu.ru

Abstract. This study explores the development and application of generative artificial intelligence (AI) for technological trend analysis and patent research automation. The research addresses the inefficiencies in traditional patent research, which is labour-intensive and costly, proposing a solution based on AI technologies such as machine learning, natural language processing (NLP), and vector database systems. The proposed solution incorporates MLOps and RAG frameworks for data collection, analysis, and integration, enabling the automation of patent data processing and keyword extraction through modified TF-IDF algorithms and semantic embeddings. The architecture includes tools for clustering patents by thematic and contextual similarities, significantly reducing the time required for research and enhancing accuracy. Experimental results demonstrate the effectiveness of the developed system, achieving significant improvements in the speed of generating patent studies (30–60 minutes) and the precision of information retrieval. The study highlights the transformative potential of generative AI in streamlining intellectual property analysis and fostering technological innovation.

Keywords: patent research automation, Generative Artificial Intelligence (GAN), Natural Language Processing (NLP), Machine Learning Operationalization (MLOps), Retrieval-Augmented Generation (RAG), TF-IDF Algorithm, Large Language Models (LLMs), vector databases

Citation: Pochetniy V.A. Integrating generative AI for technological trend analysis and patent research automation. Technoeconomics. 2025. 4. 2 (13). 4–20. DOI: https://doi.org/10.57809/2025.4.2.13.1

Научная статья
УДК 330.47
DOI: https://doi.org/10.57809/2025.4.2.13.1

ИНТЕГРАЦИЯ ГЕНЕРАТИВНОГО ИИ ДЛЯ АНАЛИЗА ТЕХНОЛОГИЧЕСКИХ ТРЕНДОВ И АВТОМАТИЗАЦИИ ПАТЕНТНЫХ ИССЛЕДОВАНИЙ

Василий Почетный ✉

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

✉ pochetnyj.va@edu.spbstu.ru

Аннотация. В данной работе исследуются методы интеграции генеративного искусственного интеллекта (ИИ) для анализа технологических трендов и автоматизации патентных исследований. Основное внимание уделено разработке программного продукта, основанного на технологиях машинного обучения, анализа естественного языка (NLP) и векторных баз данных, что позволяет автоматизировать обработку патентных данных и выделение ключевых слов. Предложенная система включает использование MLOps и RAG для автоматизации поиска, анализа и кластеризации данных. Эксперименты продемонстрировали эффективность модифицированного алгоритма TF-IDF для извлечения ключевых слов и применения семантических эмбеддингов для улучшения точности анализа. Разработанная система позволяет уменьшить время генерации патентных исследований до 30—60 минут, значительно повышая производительность и точность. В перспективе рассматривается расширение возможностей системы через интеграцию дополнительных патентных баз и прогнозирование технологических трендов с помощью ИИ.

Ключевые слова: автоматизация патентных исследований, генеративный искусственный интеллект, анализ естественного языка (NLP), MLOps, Системы RAG, алгоритм TF-IDF, большие языковые модели (LLM), векторные базы данных

Для цитирования: Почетный В.А. Интеграция генеративного ИИ для анализа технологических трендов и автоматизации патентных исследований // Техноэкономика. 2025. Т. 4, № 2 (13). С. 4—20. DOI: https://doi.org/10.57809/2025.4.2.13.1

Introduction

In the rapidly advancing digital age, companies are actively exploring new approaches to enhance efficiency, improve customer engagement, and strengthen their market positions. Generative artificial intelligence represents a groundbreaking technology that offers unique capabilities for creating, analyzing, and optimizing data in ways that were previously unattainable via traditional methods.

Generative AI impacts businesses in areas such as content detection, creation, authenticity, and regulation. It also has the potential to automate human labour and enhance interactions with both customers and employees. Key technologies in this domain include artificial general intelligence (AGI), AI engineering, autonomous systems, cloud-based AI services, composite AI, computer vision, data-driven AI, edge AI, intelligent applications, model operationalization (ModelOps), operational AI systems (OAISys), prompt engineering, smart robots, and synthetic data (Gupta, 2023).

Research on generative AI is accelerating due to the popularity of technologies like Stable Diffusion, Midjourney, ChatGPT, and large language models.

Critical technologies in this field include AI simulation, AI Trust, Risk, and Security Management (AI TRiSM), causal AI, Data Labelling and Annotation (DL&A), First Principles AI (FPAI), also known as physics-informed AI, foundation models, knowledge graphs, multi-agent systems (MAS), neuro-symbolic AI, and responsible AI. Figure 1 illustrates the placement of AI technologies on a hype cycle chart, reflecting their level of expectations and the estimated time remaining until they reach the plateau of productivity.

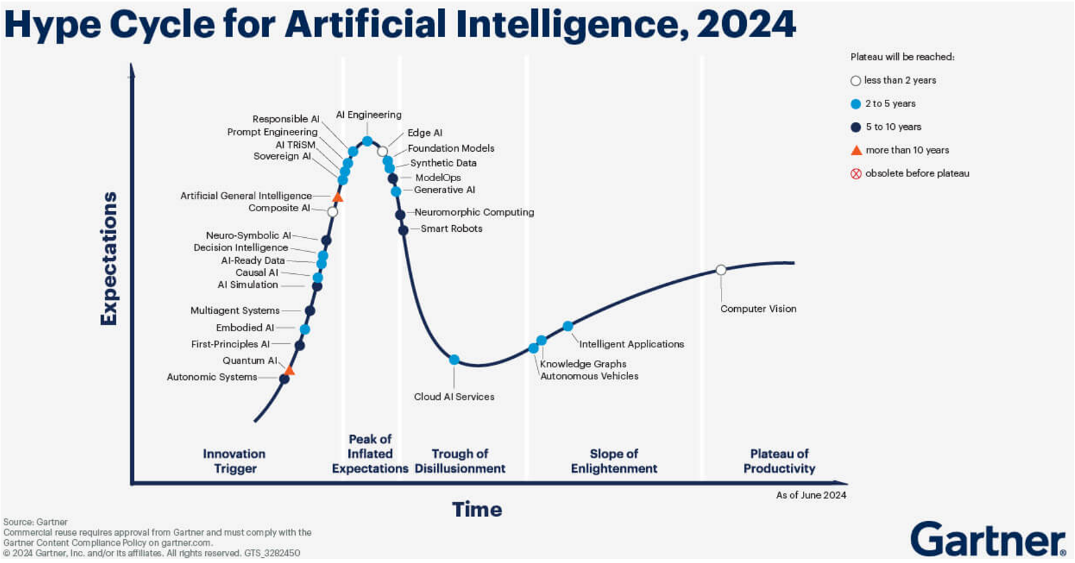


Fig. 1. The "Hype Cycle" for Artificial Intelligence in 2024 according to Gartner (Explore Beyond GenAI on the 2024. Hype Cycle for Artificial Intelligence).

Generative artificial intelligence represents a significant leap in technological advancement, offering immense opportunities across various sectors. It fosters unprecedented creativity and innovation, enabling the exploration of new ideas and solutions. Through automation and optimization, generative AI enhances efficiency, reduces costs, and frees up human resources for strategic tasks. It also delivers personalized experiences, improving customer satisfaction and strengthening relationships. By analyzing vast amounts of data, generative AI provides valuable insights and facilitates informed decision-making (Iyer, 2024; Kutuzova, 2024).

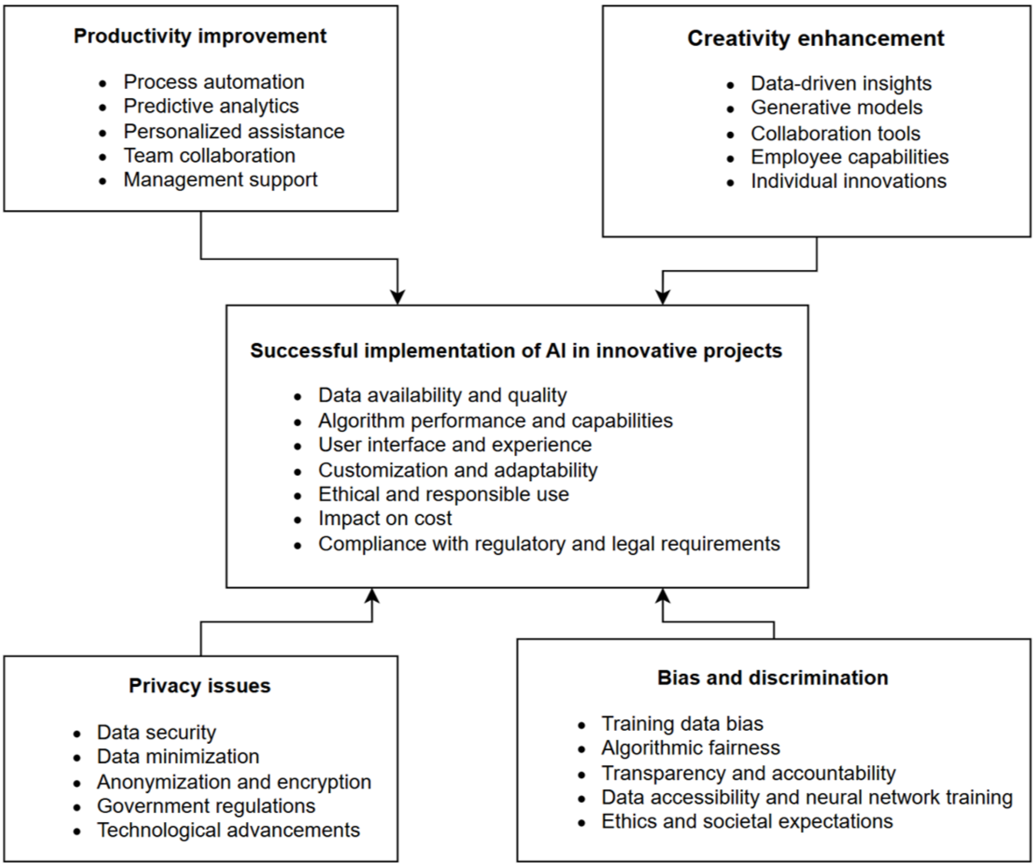
Generative AI holds outstanding potential for transforming businesses. A Gartner study conducted in 2024 revealed that among over 2.500 executives, 38% view it as a means to boost customer experience and loyalty. Additionally, 26% associate it with revenue growth, 17% with cost optimization, and 7% with ensuring business continuity (Bieck, 2024).

The primary areas of application for generative AI include marketing (14%), sales (12%), automated customer support (16%), creative tasks, and employee productivity enhancement. This technology can also significantly accelerate software development by assisting specialists in coding more efficiently. Gartner predicts that by 2027, approximately 30% of manufacturing companies will actively use generative AI to optimize product development processes.

Generative artificial intelligence has revolutionized the healthcare sector, improving diagnosis, treatment, and personalized medicine. A balanced approach to implementing generative AI across various industries, from personalized retail services to healthcare, involves investing in research, fostering collaboration, promoting digital literacy, and prioritizing ethical considerations. Future advancements include enhanced personalization, autonomous decision-making, collaborative intelligence, ethical governance, innovation, sustainability, and continuous learning (Yikilmaz, 2023).

Generative AI models, such as neural networks based on deep learning, can create new content, including images, music, and text. AI algorithms can analyze historical data for predictions and recommendations, enabling businesses to anticipate demand, optimize inventory, and efficiently allocate resources.

Successful implementation of AI in innovative and creative projects depends significantly on several key factors, including the creation of a synergistic environment where AI can maximize creativity and achieve project success, addressing privacy concerns, improving productivity, and resolving issues of bias and discrimination. Figure 2 shows a conceptual model of successful AI implementation.



Productivity improvement
Process automation
Predictive analytics
Personalized assistance
Team collaboration
Management support
Creativity enhancement
Data-driven insights
Generative models
Collaboration tools
Employee capabilities
Individual innovations
Successful implementation of AI in innovative projects
Data availability and quality
Algorithm performance and capabilities
User interface and experience
Customization and adaptability
Ethical and responsible use
Impact on cost
Compliance with regulatory and legal requirements
Privacy issues
Data security
Data minimization
Anonymization and encryption
Government regulations
Technological advancements
Bias and discrimination
Training data bias
Algorithmic fairness
Transparency and accountability
Data accessibility and neural network training
Ethics and societal expectations

Fig. 2. A conceptual model of successful AI implementation.

The purpose of this study is to explore and develop methods for integrating solutions based on generative artificial intelligence (AI) to analyze technological trends and create tools capable of effectively processing large volumes of data on technological developments and predicting future trends.

In the context of rapid technological development, businesses must continuously monitor and analyze technological trends to remain competitive in the market. Generative AI provides new opportunities for data analysis and the identification of complex patterns, making it a powerful tool for analyzing technological trends. Integrating generative AI solutions for analyzing technological trends can significantly improve the quality and speed of analysis while also offering new insights for business development. With the ability to forecast future technological trends, businesses can make more informed decisions regarding their development strategies and investment in innovation.

Thus, the integration of generative AI solutions for analyzing technological trends is highly relevant and can bring substantial benefits to companies striving to maintain leadership in their respective industries.

There is a notable problem related to the labour-intensive and time-consuming nature of patent research, which requires significant time and financial resources. Conducting a single patent study typically takes 25—30 working days, amounting to an average of 200 to 240 man-hours. Of this, 10 to 20 working days are spent on finding and organizing patent information. At the time of writing this article, the minimum cost of a patent study conducted by patent agencies is approximately 60.000 rubles. In Russia, over 50.000 patent applications were filed during 2023—2024, while the number of annual patent studies exceeds 250.000. Globally, this number amounts to approximately 3.45 million.

Patent research is the process of searching for and analyzing patent information to obtain various types of data related to patents and intellectual property. Patent research is conducted for a variety of purposes, including determining the novelty of an invention, identifying possible patent rights infringements, assessing the competitive landscape, and supporting innovation processes.

The following types of patent research are distinguished:

1. Novelty Search. This involves verifying whether an invention is new and has not been previously described in patent literature or other sources. This type of research helps to determine whether an invention can be patented.

2. Freedom-to-Operate (FTO) Search. This includes analyzing existing patents to determine whether a product or process would infringe on the patent rights of third parties. This helps companies avoid litigation and licensing issues.

3. Competitive Intelligence. This involves studying the patent activity of competitors to understand their strategies and technological directions. It allows companies to remain competitive in the market.

4. Invalidity Search. This involves searching for information that could invalidate an existing patent, such as prior patents or publications describing the same technology.

5. Patent Landscape Analysis. This is a large-scale analysis of patent information aimed at identifying technological trends, patent clusters, and market opportunities.

The goals of patent research include:

— Identification of novelty: helping to determine whether an invention is new and deserving of patent protection.

— Avoiding infringements: assisting in avoiding violations of other companies' patent rights and potential lawsuits.

— Strategic planning: enabling companies to plan their research and commercial strategies based on the patent activities of competitors.

— Evaluation of patent portfolios: assisting in assessing the value and utility of a company's patent portfolio.

Patent research is often performed using specialized databases and software such as Google Patents, Derwent World Patents Index, Espacenet, XLSCOUT, and DorothyAI. These tools help to automate the process of searching and analyzing patents, making it faster and more accurate.

Current methods for conducting patent research can be divided into two main groups: patent bureaus, where research is conducted manually by professionals, and tools utilizing neural networks, including large language models. The first group includes Garant, Guardium, and Patentus; the second group includes NLPatent, DorothyAI, and XLSCOUT (XLSCOUT About Us).

Materials and Methods

A solution to the described problem could be a software product based on the results of scientific and technical research in the fields of big data analysis, machine learning, natural language processing (NLP), and methods for solving complex data search and analysis tasks using artificial intelligence. At its core, the product leverages natural language analysis methods, including NLP models, text vector classification methods, and web scraping technology to automate the process of acquiring patent data. These tools enable patent departments and intellectual property specialists to automatically generate patent research based on data categorized by the International Patent Classification (IPC) and keywords, using natural language and machine learning methods. The development of this solution assumes the use of the technologies described below.

MLOps is a business model developed by organizations engaged in machine learning. The concept of Machine Learning Model Operationalization Management (MLOps) replaces traditional vertical structures within organizations by promoting the shared use of resources and expertise across departments. MLOps provides a way for data specialists and operational experts to collaborate and communicate effectively to manage the lifecycle of machine learning (ML) production. It is a culture and practice in machine learning engineering that aims to integrate the creation and operation of ML systems (Ops).

The main components or principles of MLOps are presented in Figure 3:

1. Automation—workflow stages can be easily automated without manual intervention.

2. CI/CD—MLOps includes continuous integration/continuous delivery (CI/CD), testing, and monitoring.

3. Version control—the ability to track ML models and datasets using version control systems.

4. Experiment tracking—allows multiple model training experiments to be run in parallel.

5. Testing—testing of various features, data, models, infrastructure, and more.

6. Monitoring—collecting detailed information on the performance of models.

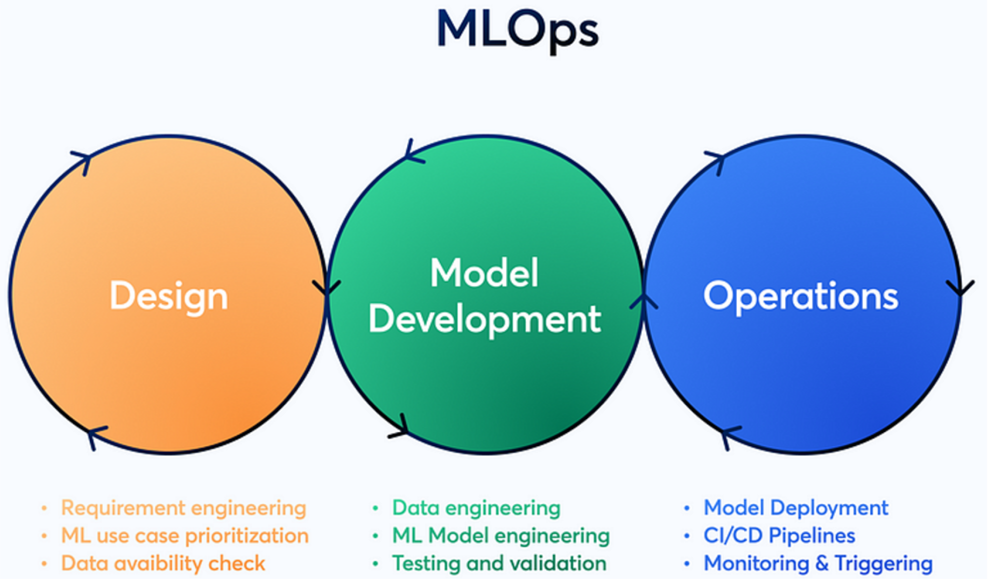


Fig. 3. MLOps principles.

The typical lifecycle or workflow of MLOps includes the following stages: data collection, data analysis, data preparation, model training, model evaluation, model validation, deployment, and monitoring (What is MLOps).

MLOps can be implemented in three different ways. The first level of implementation is a manual process, shown in Figure 4. This approach is common for companies that are just beginning to work with machine learning. A manual ML workflow may be effective if models are rarely modified or retrained.

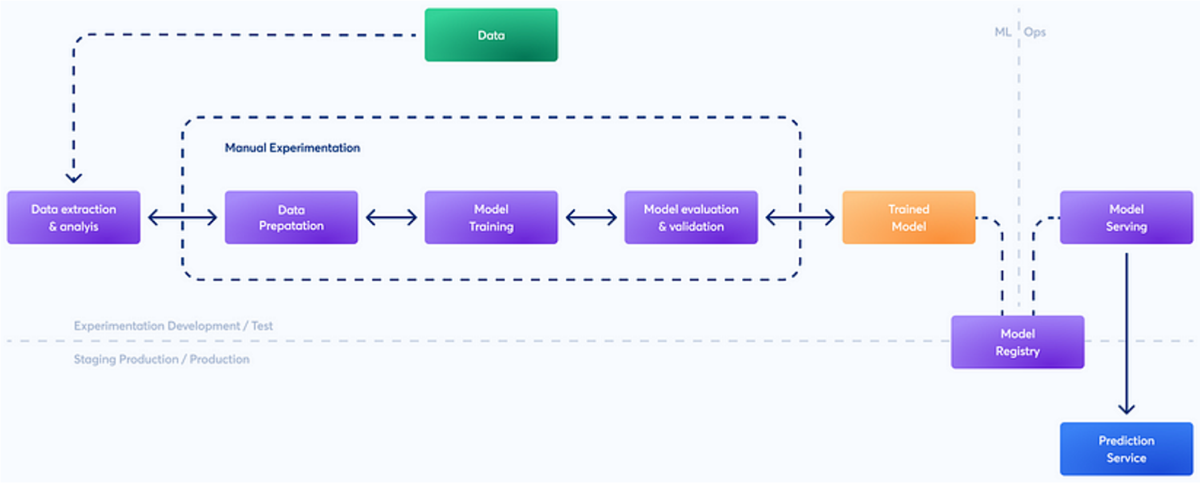

Data
ML
Ops
Manual Experimentation
Data extraction & analyis
Data Prepatation
Model Training
Model evaluation & validation
Trained Model
Model Serving
Model Registry
Experimentation Development / Test
Staging Production / Production
Prediction Service

Fig. 4. Steps of the manual MLOps implementation process.

The second level of implementation is ML pipeline automation, shown in Figure 5. This architecture is ideal for deploying new models based on fresh data rather than new machine learning concepts. It automates the ML pipeline, resulting in faster experimentation.

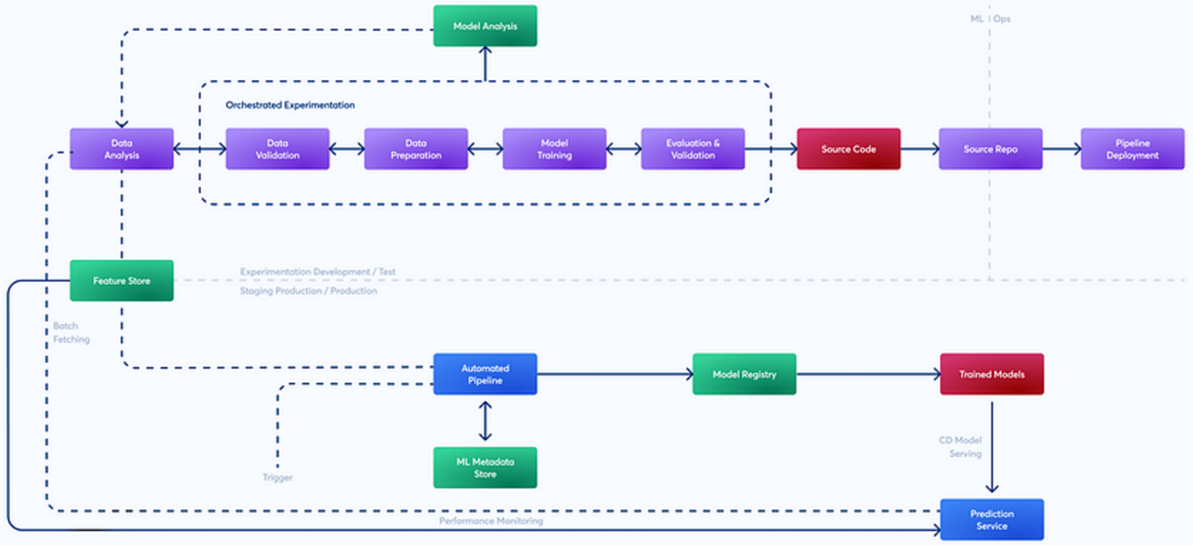


Fig. 5. ML pipeline automation.

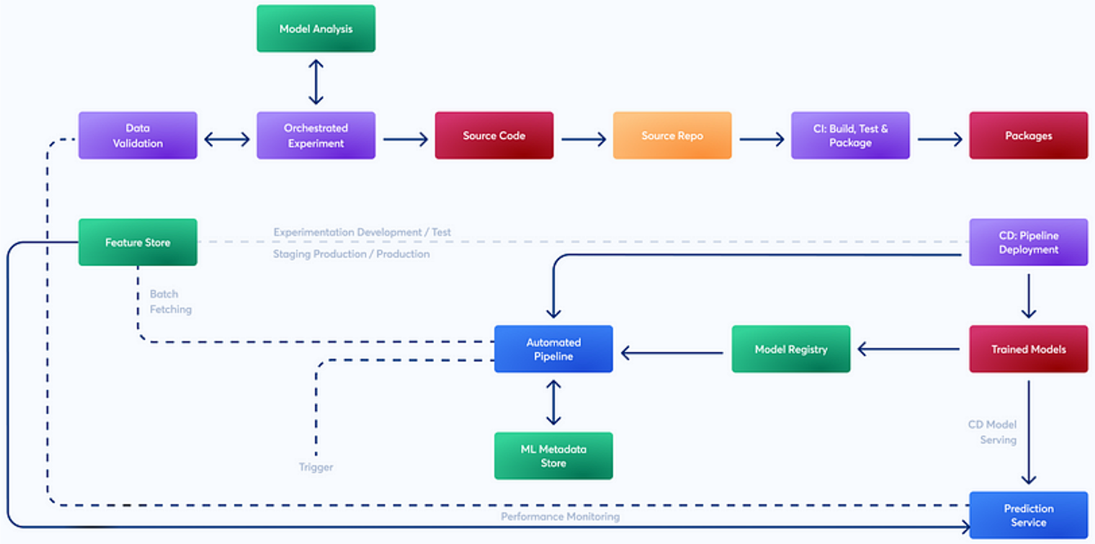


Fig. 6. Automation of the CI/CD pipeline.

MLOps enables companies to significantly reduce the time required for data analysis by creating automated feedback loops capable of identifying patterns in vast datasets without human intervention.

There are several comprehensive and specialized MLOps tools available. Comprehensive solutions for MLOps are fully managed services designed for rapid creation, training, and deployment of machine learning models, such as Amazon SageMaker and Google Cloud MLOps.

While comprehensive solutions are a good option, splitting the MLOps pipeline into multiple microservices allows organizations to build their own MLOps tool stack. These platforms are particularly well-suited for companies just starting out with machine learning: MLFlow, Neptune.ai, Weights & Biases, Cortex, and Polyaxon.

The corporate AI strategy of every organization revolves around ModelOps. ModelOps is a system that enables the integration of multiple AI objects, solutions, and frameworks while maintaining scalability and control. The sequence of ModelOps actions is shown in Figure 7.

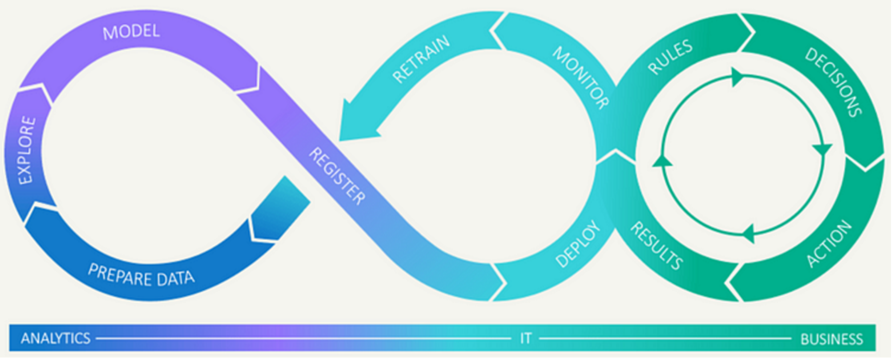


Fig. 7. ModelOps.

To put it simple, ModelOps is an extension of MLOps. In an organization wishing to implement ModelOps, MLOps must first be deployed. ModelOps requires the same skills as MLOps, plus a few additional skills related to IT operations, risk management, and some others (MLOps vs. DevOps vs. ModelOps).

ModelOps is used by companies to address issues such as:

– Model quantity—to account for differences in business processes, customization, and specialized client groups, each organization needs to manage hundreds of models.

– Complexity—even the most experienced IT teams face overwhelming complexity due to data and innovations in analytics.

– Compliance with regulations—as AI usage increases in markets, adhering to strict and constantly growing models becomes harder.

– Isolated environment—multiple teams are involved in model creation, from deployment to monitoring. Scaling AI can be challenging due to ineffective coordination between teams. ModelOps helps create an environment where models can easily move from the data science team to the IT production team.

The tasks of ModelOps are largely similar to those of MLOps. Typically, ModelOps involves working on CI/CD, development environments, testing, version control for models, and model repositories. ModelOps serves as the link between all other elements of the AI pipeline. The best tools and platforms for ModelOps, focused on models, are ModelOp, Modzy, and Datatron.

MLOps refers to the operationalization and management of AI models in production systems, while ModelOps is considered a superset of MLOps. ModelOps has an advantage over MLOps in that MLOps focuses only on machine learning models, whereas ModelOps aims to operationalize all AI models (ModelOps, MLOps, and Finding Value in Analytics).

MLOps should not be confused with DataOps, a data science domain that primarily focuses on data pipelines, providing valuable insights by connecting disparate data sources and having flexible workflows with data on a scale. DataOps is the practice of moving data operations into an automated, repeatable, scalable environment. It is a software engineering discipline that ensures high-performance data management for analytics.

MLOps is not the same as AIOps, as AIOps automates processes in the organization's IT department, not the machine learning team. AIOps—short for Artificial Intelligence Operations—is used for automating processes or detecting patterns that would not be visible to humans. AIOps uses AI to analyze data and then optimize operations based on this data.

RAG is a method for working with large language models where the user asks questions to the program, and additional information from external sources is appended, all of which is then fed into the language model. In other words, supplementary information is added to the context of the language model's request, enabling it to provide a more complete and accurate answer. Retrieval is the process of searching and extracting relevant information. The system responsible for this is called a retriever. Retrieval Augmented Generation (RAG) is a method of generating a response for the user with the additional found information in mind.

It is expensive and inefficient to retrain the model on the customer's knowledge base, such as the patent database, as this would need to be done after each change to the knowledge base. In the RAG system, it is necessary to find the relevant article in the knowledge base and provide the LLM with not only the user's question but also the relevant portion of the knowledge base to form a correct response. (Getting Started with Large Language Models)

It is costly and inefficient to further train a model on the client's knowledge base, in this case, on a patent database, because this would need to be done after every change to the knowledge base. In a RAG system, it is necessary to find the relevant article in the knowledge base and provide not only the user's question but also the relevant part of the knowledge base for generating a correct answer.

The RAG algorithm works as follows. The entire knowledge base is divided into small text fragments called chunks, the size of which can range from several lines to several paragraphs (approximately from 100 to 1000 words). Chunks are digitized using an embedder and transformed into embeddings, which are vectors. These numbers contain the hidden meaning of each chunk, allowing for semantic search. Then, the vectors are stored in a special database where they can be searched for similarity to a query.

When the user sends a query to the LLM, the query text is encoded into an embedding by the same embedder, and a search is performed in the database to find the most semantically similar vectors. Typically, cosine similarity is calculated between the query vector and chunk vectors, after which the top N most similar vectors are selected (Ahadh, 2021).

At the next stage, the text fragments corresponding to the found vectors are combined with the user's query into a single context and passed to the language model. Thus, the model "thinks" that the user not only asked a question but also provided data for the answer.

An important factor is choosing the optimal number and size of fragments. If too much unnecessary information is provided, or too little, the model will not be able to give the correct answer. The smaller the fragment, the more precise the literal search will be. The larger the fragment, the closer the search is to the semantic one. Different queries require different amounts of fragments. The optimal fragment size must be determined empirically, below which information loses meaning and clutters the context. Fragments should overlap with each other so that input to the model is continuous, not isolated pieces. The beginning and end of a fragment should make sense, ideally matching the start and end of sentences or paragraphs (Thiyagarajan, 2021).

Semantic search via embeddings does not always yield the desired results, especially for specific terms. A combined approach with TF-IDF and ranking algorithms like BM25 is often used (Okada, 2021). To improve accuracy, the user's query can be rephrased several times with the help of an LLM, and fragments can be searched using all variations. Usually, 3—5 variations of the query are made, and the results are then combined. If a lot of information is found in response to a query and it does not fit in the context, it can be simplified using the LLM and passed in a compressed form to be used in generating the answer.

The key element in the algorithm for automatic keyword extraction is the ability to provide users with fast access to the relevant content. Let's look at two common algorithms: TF-IDF (Term Frequency-Inverse Document Frequency)—a statistical measure used to evaluate the importance of a word in the context of a document, which is part of a collection of documents or corpus, literally the inverse frequency of term usage in the document—and TextRank—an algorithm for building a graph model based on the original natural language text (Tixier, 2016).

The TF-IDF algorithm was modified by adding an algorithm to calculate the weight of a patent title based on the characteristics of the patent text. Additionally, an automatic extraction algorithm using Word2Vec was applied to extract semantics.

Experimental results showed that as the number of automatically extracted keywords increased, the accuracy of the TF-IDF, TextRank, and modified TF-IDF algorithms gradually decreased, while the memorization speed increased, thus proving the effectiveness of the modified TF-IDF algorithm for automatic keyword extraction from patent texts.

The TF-IDF algorithm is a time-tested and commonly used algorithm for automatic keyword extraction. This algorithm takes into account that the significance of a word is directly proportional to how often it appears in a document but inversely proportional to how often it appears in the text corpus. The abbreviation TF refers to term frequency. The method of calculating TF is as follows:

$$TF_{i,j} = \frac{N_{i,j}}{\sum_k N_{k,j}} \qquad (1)$$

Where $N_{i,j}$, denotes the frequency of occurrence of the word i in the text d_j, and k is the number of different words in the text.

In other words:

$$TF(t,d) = \frac{Total\ number\ of\ terms\ in\ the\ document\ d}{The\ number\ of\ mentions\ of\ the\ term\ t\ in\ the document\ d}$$

The abbreviation IDF refers to the inverse frequency of use of the document.

$$IDF_i = \log \frac{|D|}{|j : t_i \in d_j|} \tag{2}$$

where $|D|$ is the total number of texts in the corpus, and $|j : t_i \in d_j|$ is the number of texts containing the word i in the corpus.

The TF-IDF value is obtained by

$$TF - IDF_{(t,d,D)} = TF(t,d) \times IDF(t,D) \tag{3}$$

If a word has a high TF value and a low IDF value, it is considered to have high criticality. This method is easy to use and widely applied.

The TextRank algorithm is an enhanced version of the PageRank algorithm (Implementation of TextRank Algorithm Methods for Keyword Extraction). The PageRank principle is that if a web page is linked to many other web pages, it indicates that the web page is relatively important, which means its PageRank value is high. Each word or phrase in a text is represented as a node in a text graph, and the relationships between words are described as edges that measure the similarity between words.

PageRank is calculated using the following formula:

$$S(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{4}$$

where $S(V_i)$ is the PR-value of web page V_i, V_j is the web page associated with V_i, i.e. the incoming link, $In(V_j)$ is the set of incoming links, and $Out(V_j)$ is the number of elements in the set of links pointing to external sites for the j web page.

PR-value is one of the most effective ways to measure the return on investment (ROI) in communications, which converts the results of any PR activities, whether it's a press release, social media advertisement, into value added to the business and reputation. PR value is a calculation of the financial benefit from reaching your target audience through paid advertising, making it easy to measure and compare ROI in paid, organic, and owned media (Understanding PR value).

TextRank is a graph-based ranking algorithm. It considers sentences or words in a text as nodes in the graph, with relationships between them as edges, and determines their importance by calculating the weights between nodes. The formula for calculating the TextRank algorithm, based on PageRank, looks as follows:

$$WS(V_i) = (1-d) + d \times \sum_{V_j \in In(V_j)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \tag{5}$$

where $WS(V_j)$ is the weight of sentence *i*, the indicator W_{ji} indicates the similarity of sentences, and d is the attenuation coefficient, is usually assumed to be 0.85.

After processing the text with word segmentation and stop word removal, the TF-HF-IDF value is calculated, and individual words are represented as Word2Vec word vectors (Sumayasuhana, 2022). Stop words are words that do not carry semantic meaning in a sentence, such as conjunctions or prepositions. Additionally, special characters must also be excluded from the text. After this, the semantic similarity of the processed words is calculated. Using the hierarchical clustering algorithm, a set of semantic thematic concepts is obtained, i.e., a set of words with similar semantics. Finally, the overall weight value is calculated:

$$W - score(t_i, d) = score(t_i, d) + \sum_{j=1}^{N} sim(t_i, t_j) \tag{6}$$

$$sim(t_i, t_j) = \cos\theta = \frac{e_i \times e_j}{|e_i| \times |e_j|} \tag{7}$$

where $\sum_{j=1}^{N} sim(t_i, t_j)$ refers to the sum of semantic similarities between the word t_i and other words, $sim(t_i, t_j)$ is a semantic similarity based on Word2vec between words t_i and t_j, e_i is a vector of words t_i, and e_j a vector of words t_j.

An example could be the automatic categorization of articles on news portals by topic using TF-IDF-based clustering. In social networks, this method can be used to group posts and create personalized news feeds. Furthermore, vector representation of texts based on TF-IDF can be applied in machine learning models for document classification. For example, TF-IDF can be used to classify emails for spam filtering or automate customer service by sorting customer queries into appropriate categories (Wang, 2024).

Geometrically, the binary classifier SVM can be represented as a hyperplane in the object space that effectively separates points corresponding to positive and negative instances. During the training process, SVM selects a hyperplane that maximally separates positive and negative examples, creating a gap or margin between them, which represents the distance from the hyperplane to the nearest points of both classes. These nearest points are called support vectors, and they play a key role in defining the hyperplane, while the rest of the training data does not influence the final decision. This property distinguishes SVM from other classification algorithms. One of the main advantages of SVM is its theoretically grounded approach to reducing the risk of overfitting, allowing it to perform well regardless of the feature space's dimensionality. Moreover, SVM does not require parameter tuning, as there is a theoretically grounded “default” choice of parameters that have demonstrated optimal performance in experimental validation.

Figure 8 shows a system model that can perform patent text classification and compare the capabilities of the TF-IDF and TF-RF algorithms (Harmandini, 2024).

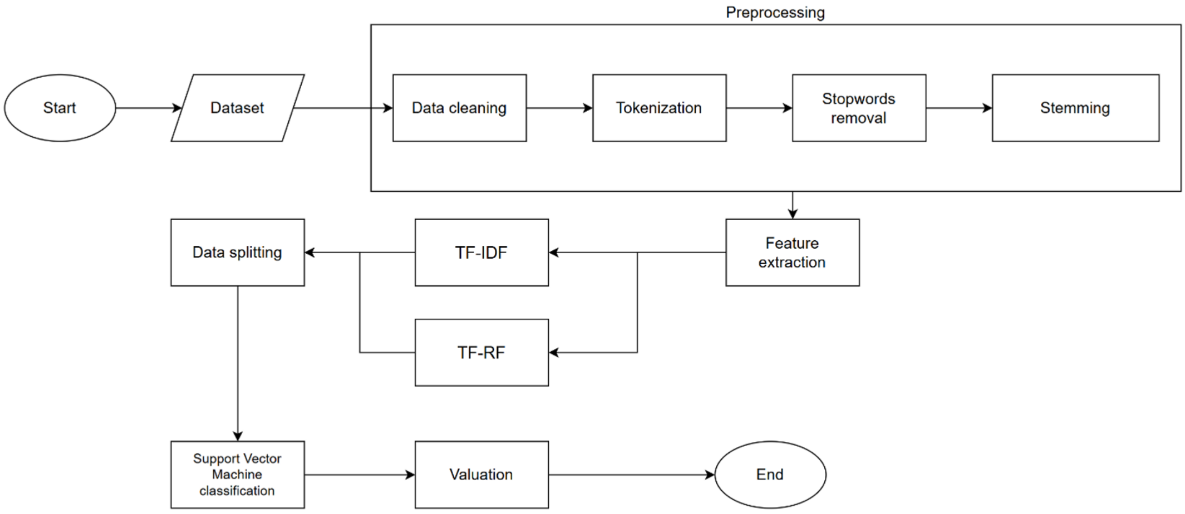


Fig. 8. A system model for performing text classification.

A comparison was made of the effect of the number of key words on the performance of the algorithm. The number of selected key words ranged from 1 to 10. For comparison of the algorithms, the following metrics were used: Precision, Recall rate, F-measure, and F1-score.

Precision is a metric that measures how accurately the model classifies positive examples among all predicted positive cases. The formula for calculating precision is a fraction or ratio where the numerator is the number of true positive examples, i.e., those cases where the model correctly predicted a positive outcome, and the denominator is the sum of true positive and false negative examples. FP (False Positive) is the number of false positive examples, where the model predicted a positive result that was actually negative:

$$Precision = \frac{TP}{TP + FP'} \tag{8}$$

Recall rate, or simply recall, is a metric used in machine learning and information retrieval that measures the ability of a model to find all relevant objects or positive examples in the data. In the context of binary classification, it is the proportion of true positive examples that the model correctly classified.

The formula for calculating recall is the ratio of the number of true positive examples to the sum of true positive and false negative results:

$$Recall = \frac{TP}{TP + FN'} \tag{9}$$

where:

TP (True Positives) is the number of examples that were correctly classified by the model.

FN (False Negatives) is the number of examples that the model incorrectly classified as negative.

Recall is important in cases where missing positive examples could have serious consequences. For example, in disease diagnosis, where it is crucial to identify all cases of illness, or in security systems where it is necessary to detect all threats.

F-measure, also sometimes called F1-score, is a metric used to evaluate the performance of machine learning models, primarily in classification tasks. It combines into a single number two important metrics, precision and recall. Thus, the F1-score is the harmonic mean of both metrics.

The formula for calculating F1-score is:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

The F1 score provides a weighted representation of how well the model performs when missing positive examples and having false positives matter. This metric is typically used when a balance between precision and recall needs to be maintained. The closer the F1-score is to 1, the better the model performs. If the F1-score is closer to 0, it indicates weak model performance.

Databases can be roughly divided into two groups: specialized vector databases and databases that support vector search. The first group includes Chroma, Vespa, Marqo, Qdrant, LanceDB, Milvus, Pinecone, and Weaviate. The second group includes OpenSearch, ClickHouse, PostgreSQL, Cassandra, Redis, Elasticsearch, Rockset, and SingleStore.

Currently, the project uses PostgreSQL, as it is an open-source product that allows storing vector embeddings alongside the original data. However, with the project's development and further scaling, there may be a transition to Milvus, as this product is well-suited for ensuring high availability and fault tolerance through built-in load balancing mechanisms. The project uses the Langchain framework, which provides faster, cheaper, and more efficient task execution compared to previous agent models (Intro to LLM Agents with Langchain).

There are three agent architectures in LangGraph that demonstrate a plan-and-execute design style (LangGraph). Building language agents as graphs. These agents promise several improvements over traditional “reflect and act” ReAct agents.

First, they can execute multi-step workflows faster because there is no need to call a larger agent after each action. Each subtask can be executed without an additional LLM call or with a lighter LLM call. Second, they offer cost savings compared to ReAct agents. If LLM calls are used for subtasks, they can usually be sent to smaller models that are domain specific. In this case, the larger model is called on only for planning and replanning steps and generating the final response.

Third, they can provide higher speed and quality of task execution, forcing the planner to explicitly consider all the steps required to complete the entire task. Generating full reasoning steps is a proven prompting method to improve results. Breaking the problem down also allows for more purposeful task execution.

A ReAct agent queries the language model using a recurring cycle of thoughts, actions, and observations. It leverages the benefits of the chain-of-thought approach, opting to choose one action at a time. While this can be effective for simple tasks, it has a couple of main drawbacks:

1. Each tool call requires an LLM call.
2. The LLM plans for only one subtask at a time. This can lead to suboptimal trajectories as it is not forced to “reason” about the entire task.

One way to overcome these two drawbacks is to perform a clear planning stage.

Plan-and-execute is a simple architecture consisting of two components:

1. A planner proposes that the LLM generate a multi-step plan to execute a large task.
2. Executors, which take the user query and a step in the plan and call one or more tools to complete that task.

Once the execution is completed, the agent is called again with a request for replanning, allowing it to decide whether to finish with an answer or generate a subsequent plan if the first plan did not yield the desired result (Plan-and-Execute Agents). This agent structure allows us to avoid calling a large LLM for planning each tool call. It is still limited by the sequential tool call and uses the LLM for each task, as it does not support variable assignments.

Results and Discussion

The architecture of the developed product is as follows. In the first stage, automated collection and systematization of data obtained from the patent registry is carried out, for example, from the Rospatent registry. Patents are stored in a PostgreSQL database consisting of 6 tables, each containing about one million rows. Then, the text data is summarized, and the classical TF-IDF method is applied to extract keywords. The vector database stores embeddings obtained using GigaChat. New patent entries from the Rospatent registry are automatically loaded into the database every 24 hours. The current information on patents in text format occupies approximately 3 TB of storage.

The user interaction with the product follows this scenario. The user makes a query on the website regarding a topic of interest. A check is performed to see if there is an existing entry in the database containing a summary of information related to the query. If no corresponding entry exists, it is added at this stage. Patent searches in the database are based on the information in the user's query and the patent title. As a result, a ranked list of the top 5, 10, or 100 results is displayed depending on the search settings, and a brief summary of the entire patent or a specific section can be obtained.

Patents are clustered together based on embeddings obtained through GigaChat. The closer the texts are in meaning, the smaller the cosine distance between them in the vector database. Patents in clusters are grouped by common topics, such as, for example, radio communications or tablet computers, or by common patent holders. The average cluster size is 100 patents. Cluster names are generated based on patent titles using GigaChat.

The product can be used for analytical work on existing solutions, tech scouting, and intellectual property protection.

In the vector database, data is stored in the form of embeddings. The user makes a query to the database, after which relevant texts are selected from the database. The retrieved texts are sorted and packaged into a prompt. Then, the top 5 or 10 most relevant results are fed into the LLM, along with the prompt, as contexts. The LLM determines the most appropriate context from the provided ones and gives a response to the user's query.

A supporting service based on LLM is used to generate complex prompts. The input to the service consists of a simple prompt and a description of the format in which the answer should be provided. The service generates a complex prompt, which is then sent to the model.

Conclusion

Among the main technical characteristics of the project, the following can be highlighted: the speed of generating patent research, the volume of content, and the accuracy of classification of the data found for analysis. The speed of generating one research study is within the range of 30–60 minutes, which is tens of times faster than manually conducting patent research.

Prospects for further product development are considered, such as the creation of personal user accounts, which will allow for the personalization of the product for companies, and the use of more patent databases, including international ones. There is also the possibility of predicting future technological trends using AI through the clustering of all patents over time and identifying corresponding patterns.

REFERENCES

Ahadh A., Binish G.V., Srinivasan R. 2021. Text mining of accident reports using semi-supervised keyword extraction and topic modeling. Process Saf Environ Prot Part B, 455(65). doi:10.1016/j.psep.2021.09.022

Bieck C., Marshall A., Dencik J. 2024. How generative AI will drive enterprise innovation. Strategy and Leadership 52(1), 23-35. doi:10.1108/SL-12-2023-0126

Gupta D., Srivastava A. 2023. The Potential of Generative AI. 338.

Harmandini K.P., Kemas M.L. 2024. Analysis of TF-IDF and TF-RF Feature Extraction on Product Review Sentiment. SinkrOn. doi:10.33395/sinkron.v8i2.13376

Iyer S.S. 2024. Light and Shadows of generative AI for individuals, organizations and Society. Biomedical Journal of Scientific & Technical Research 58(5), 50824-50820. doi: 10.26717/BJSTR.2024.58.00920

Kutuzova A. 2024. AI-support architecture in digital marketing. Technoeconomics 3, 4 (11), 69–78. DOI: https://doi.org/10.57809/2024.3.4.11.6

Okada M., Lee S.S., Hayashi Y., Aoe J., Ando K. 2021. An efficient substring search method by using delayed keyword extraction. Inf Process Manag 37, 741. doi:10.1016/S0306-4573(00)00050-9

Sumayasuhana S., Ashokkumar S. 2022. An enhancement in machine learning approaches for novel data mining serendipitous drug usage to reduce false positive rate from social media comparing word2vec Algorithm. ECS Trans 107, 13329. doi:10.1149/10701.13329ecst

Thiyagarajan G., Prasanna S., Uma B. 2021. Automation of discussion board evaluation through keyword extraction techniques: a comparative study. IOP Conference Series: Materials Science and Engineering 1131. doi:10.1088/1757-899X/1131/1/012017

Tixier A., Malliaros F., Vazirgiannis M. 2016. A graph degeneracy-based approach to keyword extraction. Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 1860–1870. doi:10.18653/v1/D16-1191

Wang Y. 2024. Research on the TF-IDF algorithm combined with semantics for automatic extraction of keywords from network news texts. Journal of Intelligent Systems 33, 20230300. doi:10.1515/jisys-2023-0300

Yikilmaz I., Halis M. 2023. Generative AI and Innovation. Conference: 8th International CEO Communication, Economics, Organization & Social Sciences Congress, 519-527.

Explore Beyond GenAI on the 2024 Hype Cycle for Artificial Intelligence. URL: https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence (accessed: 02.05.2025).

Getting Started with Large Language Models: Key Things to Know. URL: https://flyte.org/blog/getting-started-with-large-language-models-key-things-to-know#what-are-llms (accessed: 20.05.2025).

Implementation of TextRank Algorithm Methods for Keyword Extraction. URL: https://medium.com/@theofany007/implementation-of-textrank-and-methods-for-keyword-extraction-b84f8f145b2e (accessed: 18.05.2025).

Intro to LLM Agents with Langchain: When RAG is Not Enough. URL: https://towardsdatascience.com/intro-to-llm-agents-with-langchain-when-rag-is-not-enough-7d8c08145834 (accessed: 22.05.2025).

LangGraph. Building language agents as graphs. URL: https://langchain-ai.github.io/langgraph/ (accessed: 15.05.2025).

MLOps vs. DevOps vs. ModelOps. URL: https://medium.com/censius/mlops-vs-devops-vs-modelops-2f86265881fa (accessed: 01.05.2025)

ModelOps, MLOps, and Finding Value in Analytics. URL: https://odsc.medium.com/modelops-mlops-and-finding-value-in-analytics-66cacd179944 (accessed: 03.06.2025).

Plan-and-Execute Agents. URL: https://blog.langchain.dev/planning-agents/ (accessed: 23.05.2025).

Understanding PR value. URL: https://www.harveyandhugo.com/paws-for-thought/public-relations/understanding-pr-value/ (accessed: 22.05.2025).

What is MLOps. URL: https://towardsdatascience.com/what-is-mlops-everything-you-must-know-to-get-started-523f2d0b8bd8 (accessed: 02.05.2025).

Xlscout About Us. URL: https://xlscout.ai/about-xlscout (accessed: 07.05.2025).

СПИСОК ИСТОЧНИКОВ

Ahadh A., Binish G.V., Srinivasan R. 2021. Text mining of accident reports using semi-supervised keyword extraction and topic modeling. Process Saf Environ Prot Part B, 455(65). doi:10.1016/j.psep.2021.09.022

Bieck C., Marshall A., Dencik J. 2024. How generative AI will drive enterprise innovation. Strategy and Leadership 52(1), 23-35. doi:10.1108/SL-12-2023-0126

Gupta D., Srivastava A. 2023. The Potential of Generative AI. 338.

Harmandini K.P., Kemas M.L. 2024. Analysis of TF-IDF and TF-RF Feature Extraction on Product Review Sentiment. SinkrOn. doi:10.33395/sinkron.v8i2.13376

Iyer S.S. 2024. Light and Shadows of generative AI for individuals, organizations and Society. Biomedical Journal of Scientific & Technical Research 58(5), 50824-50820. doi: 10.26717/BJSTR.2024.58.00920

Kutuzova A. 2024. AI-support architecture in digital marketing. Technoeconomics 3, 4 (11), 69—78. DOI: https://doi.org/10.57809/2024.3.4.11.6

Okada M., Lee S.S., Hayashi Y., Aoe J., Ando K. 2021. An efficient substring search method by using delayed keyword extraction. Inf Process Manag 37, 741. doi:10.1016/S0306-4573(00)00050-9

Sumayasuhana S., Ashokkumar S. 2022. An enhancement in machine learning approaches for novel data mining serendipitous drug usage to reduce false positive rate from social media comparing word2vec Algorithm. ECS Trans 107, 13329. doi:10.1149/10701.13329ecst

Thiyagarajan G., Prasanna S., Uma B. 2021. Automation of discussion board evaluation through keyword extraction techniques: a comparative study. IOP Conference Series: Materials Science and Engineering 1131. doi:10.1088/1757-899X/1131/1/012017

Tixier A., Malliaros F., Vazirgiannis M. 2016. A graph degeneracy-based approach to keyword extraction. Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 1860–1870. doi:10.18653/v1/D16-1191

Wang Y. 2024. Research on the TF-IDF algorithm combined with semantics for automatic extraction of keywords from network news texts. Journal of Intelligent Systems 33, 20230300. doi:10.1515/jisys-2023-0300

Yikilmaz I., Halis M. 2023. Generative AI and Innovation. Conference: 8th International CEO Communication, Economics, Organization & Social Sciences Congress, 519-527.

Возмозможности GenAI в рамках цикла рекламы искусственного интеллекта 2024 года. URL: https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence (дата обращения: 02.05.2025).

Начало работы с большими языковыми моделями: ключевые моменты, которые нужно знать. URL: https://flyte.org/blog/getting-started-with-large-language-models-key-things-to-know#what-are-llms (дата обращения: 20.05.2025).

Реализация методов алгоритма TextRank для извлечения ключевых слов. URL: https://medium.com/@theofany007/implementation-of-textrank-and-methods-for-keyword-extraction-b84f8f145b2e (дата обращения: 18.05.2025).

Введение к LLM-агентам с Langchain: Когда RAG недостаточно. URL: https://towardsdatascience.com/intro-to-llm-agents-with-langchain-when-rag-is-not-enough-7d8c08145834 (дата обращения: 22.05.2025).

LangGraph. Построение языковых агентов. URL: https://langchain-ai.github.io/langgraph/ (дата обращения: 15.05.2025).

MLOps VS DevOps VS ModelOps. URL: https://medium.com/censius/mlops-vs-devops-vs-modelops-2f86265881fa (дата обращения: 01.05.2025).

ModelOps, MLOps и поиск ценности в аналитике. URL: https://odsc.medium.com/modelops-mlops-and-finding-value-in-analytics-66cacd179944 (дата обращения: 03.06.2025).

Агенты по планированию и исполнению. URL: https://blog.langchain.dev/planning-agents/ (дата обращения: 23.05.2025).

Понимание ценности PR. URL: https://www.harveyandhugo.com/paws-for-thought/public-relations/understanding-pr-value/ (дата обращения: 22.05.2025).

Что такое MLOps. URL: https://towardsdatascience.com/what-is-mlops-everything-you-must-know-to-get-started-523f2d0b8bd8 (дата обращения: 02.05.2025).

Xlscout. Подробнее о нас. URL: https://xlscout.ai/about-xlscout (дата обращения: 07.05.2025).

INFORMATION ABOUT AUTHOR / ИНФОРМАЦИЯ ОБ АВТОРЕ

POCHETNIY Vasiliy A. – student.
E-mail: pochetnyj.va@edu.spbstu.ru
ПОЧЕТНЫЙ Василий Антонович – студент.
E-mail: pochetnyj.va@edu.spbstu.ru

Статья поступила в редакцию 10.06.2025; одобрена после рецензирования 16.06.2025; принята к публикации 18.06.2025.

The article was submitted 10.06.2025; approved after reviewing 16.06.2025; accepted for publication 18.06.2025.