International Journal of Innovative Research and Scientific Studies, 8(4) 2025, pages: 491-499

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Text analytics methods for automatic annotation of scientific documents

Adilbek Tanirbergenov[1], Madi Akhmetzhanov[2], Zhazira Taszhurekova[3], Munaram Khassanova[4], Bolat Tassuov[5*]

[1]*L.N. Gumilyov Eurasian National University, Astana, Republic of Kazakhstan.*
[2,3,4,5]*Taraz University named after M.Kh. Dulaty, Taraz, Republic of Kazakhstan.*

Corresponding author: Bolat Tassuov (*Email: b.tasuov@dulaty.kz*)

Abstract

This study aims to develop a hybrid system for the automatic annotation of scientific texts that efficiently processes multilingual publications using state-of-the-art natural language processing (NLP) technologies. The system integrates classical algorithms (Gensim, NLTK) with transformer-based models via the Cohere API to achieve high semantic consistency and accuracy in annotations. The system architecture comprises modules for data acquisition, preprocessing, manual and automatic annotation, data storage, and quality control. The performance of the proposed model was benchmarked against established methods such as BERTSUM, TF-IDF + LSA, and GPT-3.5-turbo using evaluation metrics including ROUGE, BLEU, and METEOR. The hybrid model outperformed other automated systems, demonstrating superior scores across ROUGE-1 (0.52), BLEU (0.41), and METEOR (0.39) metrics, indicating its effectiveness in producing concise and semantically accurate summaries. The system also achieved 100% language detection accuracy and 90% accuracy in semantic word relationships via Word2Vec. The integration of traditional statistical methods with advanced transformer models enables the proposed system to deliver high-quality annotations suitable for diverse scientific domains. The results validate the model's ability to process and summarize complex scientific texts effectively. This system provides a scalable, secure, and user-friendly platform for researchers, institutions, and developers. It supports multilingual annotation, seamless API integration, and potential deployment in cloud environments, offering significant benefits for academic, biomedical, and information-intensive sectors.

Keywords: Automatic summarization, BLEU, Cohere API, Gensim, Machine learning, METEOR, Multitasking models, NLP, ROUGE, Scientific articles, Text annotation.

DOI: 10.53894/ijirss.v8i4.7876
Funding: This study received no specific financial support.
History: Received: 2 May 2025 / **Revised:** 4 June 2025 / **Accepted:** 6 June 2025 / **Published:** 18 June 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

The annual increase in the volume of scientific research and academic papers makes it necessary to process information and communicate it more efficiently. Obtaining automatic annotations from scientific articles is one of the solutions to this problem. This method helps to present a large amount of scientific information in a concise, understandable form, which is especially important for researchers, students, and the scientific community. Traditionally, writing annotations by hand takes a lot of time and effort, and the use of artificial intelligence and Natural Language Processing (NLP) methods allows for the automation of this process. In this regard, the development of a system for automatically extracting annotations from scientific articles is one of the most pressing problems today.

Currently, the extraction of important information from the text, its structuring, and presentation in an understandable form is one of the main directions of the NLP industry. Various methods are used in this area, including machine learning, semantic analysis, vectorization, and neural networks. With the help of such methods, it is possible to improve the quality of obtaining annotations. As part of this study, we aim to create an effective hybrid model using the Cohere API, NLTK, and Gensim libraries.

Most scientific articles have a complex structure; their content is divided into different sections, and semantic and syntactic analysis of the text is required to highlight the main ideas. To this end, the effectiveness of creating an automated annotation system using modern methods for processing texts is studied. To obtain an annotation, it is necessary to use not only traditional statistical methods but also transformer-based language models because they are able to understand the context of the text more deeply.

The purpose of the study is to develop a system for automatically obtaining annotations from scientific articles and evaluating its effectiveness. To achieve this goal, several tasks are set. First of all, it is necessary to study the methods of extracting meaningful information from the text. Secondly, it is necessary to create an effective hybrid model using the Cohere API, NLTK, and Gensim libraries. Thirdly, it is necessary to evaluate the results of the model's work according to the ROUGE metric and language accuracy indicators.

2. Methods and Materials

Modern approaches to automatic annotation of scientific documents are based on a variety of text analytics methods, including both traditional statistical algorithms and modern neural network models. Classical extractive methods such as TF-IDF and latent semantic analysis (LSA) are used to identify the most informative sentences in the text [1]. More advanced models based on neural network architectures, such as BERTSUM and SciBERTSUM, are adapted to the specifics of scientific vocabulary and the structure of articles, including annotations, introductions and conclusions [2, 3]. The works of Lee et al. [4] and Gu et al. [5] demonstrate the effectiveness of using combined extractive and abstract approaches, where key sentences are first selected, and then a concise reformulated content is formed on their basis. Additionally, the study by Yasunaga et al. [6] suggests using the discursive structure of a scientific text (Background, Methods, Results, Conclusion) to improve the quality of annotations. In the biomedical field, specialized methods are used, for example, BioBERT and PubMedBERT, which show high accuracy in extracting semantic units from medical publications [7]. Methods based on ontologies and subject thesauri are also promising, which makes it possible to ensure semantic consistency of annotations [8]. The integration of attention mechanisms and reinforcement learning, as indicated in, allows models to take into account context and target preferences when generating summary descriptions. Despite significant progress, challenges remain, including the lack of marked-up datasets and difficulties in interpreting the results of neural network models [9].

In their work, Gidiotis and Tsoumakas [10] propose an original approach to the task of summarizing scientific documents based on the principle of "divide and conquer". Their method involves splitting scientific articles into structural components such as an introduction, methods, results, and discussion, which are then independently processed using the seq2seq neural network architecture. This approach takes into account the specifics of the structure of scientific texts and avoids the information noise characteristic of the holistic processing of long documents. The authors demonstrate that the proposed method surpasses the basic models in terms of the quality of summarization, which makes it especially useful for applications in bibliographic systems and automatic abstract services [10].

In this publication, the authors are developing an automatic annotation system for scientific texts for keyword extraction tasks based on machine learning. The main focus is on creating an annotated corpus suitable for training and evaluating information extraction models. The innovation of the work consists in combining automatic and manual annotation methods with subsequent validation, which ensures high data reliability. The authors also emphasize the importance of compatibility with existing scientific standards and the potential application of the approach in various biomedical and scientific information systems [11].

The preliminary version of the article, published on arXiv, examines in depth the methods of automatic annotation of keywords in biomedical articles. Unlike the 2024 publication, it describes in detail the architecture of the system, the annotation process, as well as quality and reproducibility criteria. The research focuses on the issues of scalability and extensibility of the annotated corpus, as well as ensuring transparency in the annotation process for subsequent retraining and testing of models. This makes this work especially valuable for research teams developing scalable NLP systems for biomedical data [11].

The authors conducted a comparative analysis of natural language processing tools for automatic annotation assignment in accordance with Gene Ontology based on scientific publications. The study examined various approaches, including those based on rules, statistics, and learning, and evaluated them using accuracy, completeness, and F-measure metrics. The results obtained made it possible to identify the key strengths and weaknesses of each tool, which are especially useful for researchers seeking to choose the most appropriate tool within a specific bioinformatics project [12].

This paper presents a system for extracting knowledge from biomedical texts in order to generate semantic annotations. Both linguistic and statistical methods are used to identify terms and their semantic relationships. These annotations can then be integrated into ontologies or knowledge bases, contributing to a deeper interpretation of scientific data. The work highlights the importance of automatic annotators to support ontological modeling and knowledge extraction in the context of the rapidly growing volume of biomedical literature [13].

The analysis of modern approaches to automatic annotation of scientific texts demonstrates a clear trend toward hybridization and specialization of methods depending on the domain and the type of scientific content. While early extractive techniques like TF-IDF and LSA laid the groundwork for identifying relevant textual units, the field has since advanced through the integration of neural architectures such as BERTSUM, SciBERTSUM, and BioBERT, which are fine-tuned for handling domain-specific language and document structure. Studies like those by Gidiotis and Tsoumakas emphasize the importance of document segmentation and contextual awareness in improving summarization quality, particularly in structured scientific literature. Meanwhile, works focused on biomedical texts show the growing reliance on both annotated corpora and ontology-driven methods to enhance semantic coherence and model interpretability. These developments underline a shift toward more adaptive, context-sensitive, and semantically aware systems, although persistent challenges such as dataset scarcity and the explainability of deep models continue to constrain large-scale deployment. Thus, future progress in this area will likely depend on the creation of comprehensive, high-quality training datasets and the incorporation of transparent, interpretable mechanisms into annotation pipelines.

3. Results

Text annotation plays an important role in natural language processing (NLP) tasks, especially in building corpora, training machine learning models, and creating information retrieval systems. This paper examines the architecture of a hybrid annotation system that includes both manual and automatic data markup, as well as provides scalable access to the annotated corpus.

At the first stage, texts are received, which is implemented through web scraping and interaction with the API of external sources. For these purposes, tools such as Scrapy and BeautifulSoup are used, as well as RSS feed parsers that provide regular updates to the corpus.

The next stage is text preprocessing, which includes HTML tag removal, normalization, tokenization, and lemmatization. This part of the system uses NLTK and spaCy libraries, as well as regular expressions to remove unwanted characters and structure text. Such preparation is necessary to ensure the quality of the subsequent annotation.

After preprocessing, the system proceeds to annotation. Two approaches have been implemented: manual markup using a specially designed user interface and automatic annotation based on modern NLP algorithms. Automatic annotation includes named entity definition (NER), POS tagging, and parsing. For this purpose, models based on spaCy, BERT and other transformers, as well as tools such as Stanford NLP, are used. This allows you to significantly speed up the markup process and reduce the burden on experts.

The marked-up data is stored in a centralized annotation database based on databases such as PostgreSQL and MongoDB, providing fast access and scalability. To increase the reliability of the markup, a quality control module is used, which automatically checks annotations and calculates the degree of agreement between annotators (for example, using the Cohen coefficient). Rule-based checks based on specified patterns and constraints are also implemented.

Data access is provided through a programming interface implemented as a RESTful API using the FastAPI and Flask frameworks. This allows the system to be integrated with external applications and provides researchers with convenient data export tools.

An important part of the system is user management and authentication, which ensures security and separation of access rights. Standard authorization mechanisms are used, including OAuth 2.0, JWT, and the role-based access model (RBAC), which effectively manage user rights when working with data.

Thus, the presented annotation system combines the flexibility of manual markup, the efficiency of modern NLP algorithms, and the scalability of the server architecture. This makes it suitable for use in academic, corporate, and application projects related to text data analysis.

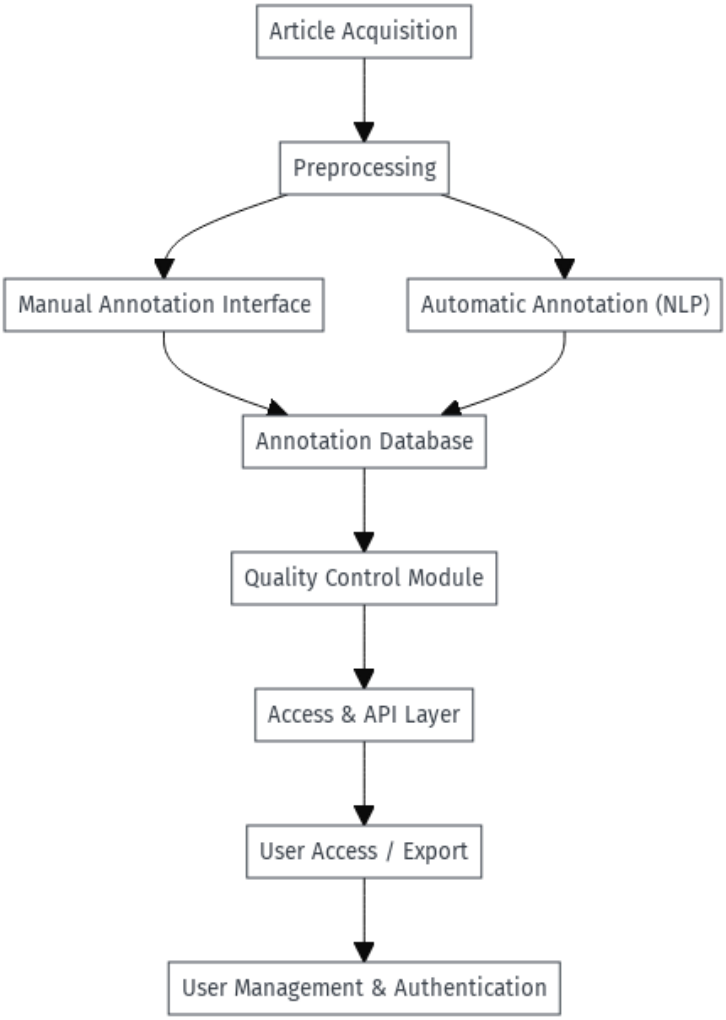

Figure 1.
Architecture of the system for automatic.

This Figure 1 represents the architecture of the system for automatic and manual annotation of scientific articles. The process begins with the Article Acquisition stage, followed by text Preprocessing, which is necessary to unify the format and clean up the data. Next, the system is divided into two parallel streams: manual annotation via a specialized interface (Manual Annotation Interface) and automatic annotation using natural language processing (NLP) methods (Automatic Annotation). The results of both streams are stored in a centralized Annotation Database, followed by a Quality Control Module that ensures the accuracy and consistency of metadata. The next steps include access to annotations via API (Access & API Layer), user data export (User Access/Export), and finally access rights and authentication management (User Management & Authentication). This modular architecture increases the flexibility, extensibility, and reliability of the entire annotation system.

The rapid development of information technology requires automation and effective processing of scientific research [14]. In this regard, the development of a system for automatically obtaining annotations from scientific articles is an important task. This system allows you to present the content of texts in a concise and meaningful way using modern methods of processing them. In addition, the system processes information from various sources and provides presentation to users through a user-friendly interface. The main purpose of the system is to receive a scientific article submitted by the user, edit it, determine the main semantic content and make a brief annotation. To achieve this goal, the system consists of several important stages. First, the process of receiving text and its primary processing is performed. At this stage, the system will clear the text of unnecessary elements and determine its structure. Second, the text undergoes semantic analysis using the Cohere API or other natural language processing (NLP) tools. Finally, the resulting annotation is presented in a user-friendly format. The system architecture consists of three main parts: front-end, back-end, and database. The front end provides a user interface that receives text from the user and displays the result. Beck-end implements basic processing processes, that is, it processes text using the Cohere API and writes the results to the database. And the database is used to store all downloaded texts and annotations and manage them. The front-end architecture is built on the basis of modern web technologies for user convenience. This interface provides for the ability to download files, track processing progress, and view results. Users can download articles in PDF or DOCX format and get their summaries. In addition, the interface provides an intuitive design and easy navigation. The beck-end system works on the Flask or Django platform. The main task of this section is to receive data received from the user, process them using the Cohere API, and store the results obtained. The beck-end system consists of several important components, including a text preprocessing module, an annotation generation module, and a database

communication module. In addition, the server takes special security measures to ensure security and stability. The database architecture is specifically designed for efficient data storage and management. The system has several main tables: users, articles, annotations, and system logs. The user table stores the data of registered users. The table of articles stores information about uploaded scientific articles, and the table of annotations records the edited texts and their summaries. The system logs table allows you to track the process of functioning of the system. The process of processing texts consists of several stages. At the first stage, the texts are cleared of unnecessary characters, that is, excess spaces and formatting errors are eliminated. At the second stage, the process of determining the language takes place, which is especially important when working with multilingual data. In the third stage, the text is analyzed using the Cohere API and a summary is created. At the last stage, the resulting annotation is displayed to the user. To improve the quality of the created annotations, the system uses certain algorithms. Annotations created based on the Cohere API contain the main content of the original text and convey information in a compact form. This system plays an important role when analyzing long texts, as it stores only the necessary information and removes excess data. As a result, the user can get the main content of a scientific article in a concise and concise form. For the convenience of the user experience of the system, its interface is designed to be simple and understandable. After the user downloads the article, the annotation will be ready in a few seconds. In addition, the system has indicators that show the progress of processing, which helps the user to understand at what stage the process is. A version of the system adapted for mobile devices is also provided, so that users can edit their data from any device. System security and data protection measures are among the most important aspects. To protect users ' personal data, the system has implemented data encryption, authentication and access control mechanisms.

These measures will help protect the system from unauthorized access and ensure data security. In addition, the system does not store personal data of users and uses it only for the processing process. The scalability of the system is also one of the most important aspects. If the number of users increases, the possibility of maintaining the operating performance of the system by increasing the server power is provided. Furthermore, the system is built on the basis of microservice architecture, which allows it to be easily expanded in the future. The possibility of integrating the system with other platforms is provided. Through the API, the system can be integrated with scientific article databases, libraries, or other information systems. It allows users to process their data on a single platform and combine information from different sources. For the commercialization of the system, its submission to universities and scientific organizations is provided. Additionally, it is planned to launch a separate platform for academic researchers. There is also the possibility of ensuring the financial stability of the system by selling an API license. You can increase the availability of the system by bringing it to the cloud. Deployment on AWS, Azure, or Google Cloud platforms ensures system stability and security. This gives users the ability to use the system online from any device. The overall architecture of the system clearly shows the interconnection between its main components. The front-end provides a user interface, the back-end processes the text, and the database stores the received data. Thus, the system effectively implements the process of automatically creating annotations from scientific articles. Figure 2 describes the architecture of the system for automatically extracting annotations from scientific articles. The main purpose of the system is to briefly and meaningfully present the main content of a scientific article. To do this, text data goes through several stages: entering a document, shortening the text, defining the language, and annotating using a neural network.

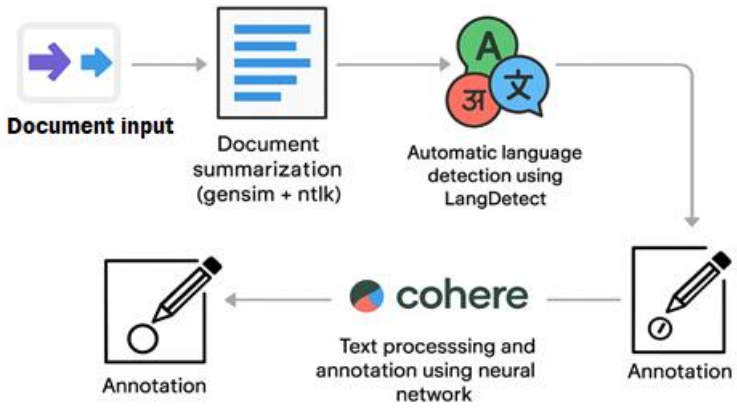

Figure 2.
General system architecture.

This Fig. 2 illustrates a more applied text annotation process that combines traditional and neural network approaches. The initial element is the input document, which first goes through the automatic summarization stage using the Gensim and NLTK (Document Summarization) libraries. Next, the text language is determined using the LangDetect (Automatic Language Detection) library, which is important when working with multilingual corpora. The information received is sent to a neural network-based model from Cohere (Text Processing and Annotation using Neural Networks), which performs automatic annotation. If necessary, manual correction or verification of annotations can be applied. This approach makes it possible to integrate classical linguistic methods with the modern capabilities of neural network models, providing high accuracy, adaptation to different languages, and potential support for multilingual research streams.

The annotation generated by the system is output as the final result. This annotation should be short, concise, and clear to the reader. The annotation fully describes the main content of the scientific article and eliminates excess information. This is especially useful for people who are engaged in scientific research because they can obtain only the main conclusions without spending time reading large texts. Each part of the image reflects a certain stage of the system, and each of them plays an important role in the text processing process. The system is fully automated; that is, the user is required only to enter the original document, and the rest of the processes are performed independently.

This system is adapted to work with multilingual scientific articles. If the user uploads an article in English, Russian, Kazakh, or Spanish, the system will automatically detect it and annotate it accordingly. This system provides the ability to adapt to many languages, not being limited to just one language. The architecture shown in the figure is based on modern methods of artificial intelligence and word processing. It is designed to process large amounts of scientific data, extract important information, and present it in a user-friendly form. This system can be a great auxiliary tool for the scientific community. In summary, this architecture reflects the process of creating automatic annotations. Each stage of the system plays an important role, and these stages work in close interaction. Such a method simplifies the analysis of scientific articles and speeds up the processing of information.

The main components of the system – model training, testing, and evaluation – play a crucial role in modern information technologies and the field of artificial intelligence. These components are closely interconnected, enabling a system to function fully and efficiently. Each component covers specific tasks at different stages, playing an essential role in determining the quality of the results. The work carried out encompasses all three components, aiming to understand and apply each one effectively.

Model training is the most fundamental stage of the system. At this stage, the system learns from actual data, meaning that algorithms and models acquire the knowledge and skills needed to perform specific tasks. The selection of training data is very important, as the system can only learn from the data provided. For example, if we train a model using texts, those texts might be from different languages, topics, or writing styles. Each data source contributes new information to the algorithms and enhances the model's representation.

During the training phase, neural networks and other machine learning methods are used. These methods are often designed for processing large datasets, as they can perform complex computations quickly. Tuning hyperparameters is also very important for effective model training, as proper tuning improves model accuracy. Factors such as the volume and quality of data and the methods used to collect the data significantly impact the training process

After training, the testing phase is needed to evaluate the model's effectiveness and accuracy. In this phase, new data that is similar to but different from the training data is used. The model makes predictions based on this test data, and the results are compared with the actual answers. This comparison helps determine how well the model works. The importance of test data lies in its ability to evaluate the model's performance in real-world conditions. One commonly used method in the testing phase is cross-validation. This method divides the data into several subsets, allowing training and testing to be performed on each group. As a result, the model is evaluated not only on a single dataset but on diverse data, providing a more comprehensive assessment. Cross-validation offers insight into the model's accuracy, stability, and overall effectiveness. It also helps evaluate how well the model performs in challenging scenarios, revealing its generalization capability. The evaluation phase is the final stage of the system. At this stage, the model's performance is assessed using various metrics. Different evaluation methods and metrics are applied to verify how well the model works. For example, metrics like ROUGE or BLEU may be used to evaluate a text summarization model. These metrics show how effectively the model performs and whether it meets user requirements.

The evaluation stage synthesizes the results of training and testing, allowing a comprehensive assessment of the model's overall performance and efficiency. It helps identify the model's strengths and weaknesses, and if necessary, guides further improvement steps. For instance, if the model's accuracy is low, it may need to be retrained or have its hyperparameters adjusted. Evaluation results can also be used to compare the model with others, helping to understand its relative effectiveness.

Table 1 presents several key metrics for evaluating the effectiveness of different components of the system. Each metric is individually important and allows for a full assessment of the system's performance.

Table 1.
Indicators and characteristics of models in the system.

Metric	Accuracy	Description
ROUGE-1	93.10%	This metric evaluates word-level matches between texts. ROUGE-1 indicates how similar the words in the generated summary are to those in the original text. A high score suggests the model produces good results.
ROUGE-L	87.60%	ROUGE-L measures the longest common subsequence between texts, reflecting structural similarity. An 87.60% score shows that the model understands the structure well, though minor differences may still exist.
Language Detection Accuracy	100.0%	This metric shows how accurately the model identifies the language of a text. A 100% score indicates perfect performance with no errors in language detection.
Word2Vec Model Accuracy	90.0%	This metric reflects how accurately the Word2Vec model detects semantic similarities between words. A 90% score indicates the model performs very well in identifying word similarities.

The ROUGE-1 and ROUGE-L metrics measure the system's performance in creating and maintaining the structure of a text summary. Indicators of 93.10% and 87.60%, in general, mean that the model has a high ability to shorten the text and convey the main idea. But, to achieve high results, there must be a complete correspondence between the words and the structure, so there is a chance that the results will improve.

When the accuracy of language detection is 100%, it shows that the system's ability to work with texts in each language is very high. This result proves that the model chose the correct language for all the inserted texts and did not make any mistakes. This indicator is important for multilingual systems because the correct understanding of features between languages directly affects the accuracy of further processing work.

The accuracy of the Word2Vec model is 90%, which indicates that the model is very good at determining semantic relationships between words. Understanding the context of words and correctly identifying their relationship to each other increases the effectiveness of the system. This result can be considered as the fact that the Word2Vec model correctly understands words and their meanings and, therefore, gives better results in word processing.

In general, the results of this table show that the System Works highly efficiently and that each component fits well. Each indicator gives a result at a high level, which indicates the quality of the work done and the model.

Figure 3 shows the values of the ROUGE-1 and ROUGE-L metrics as histograms. The histogram has two columns, one of which is indicated for ROUGE-1 and the other for ROUGE-L.

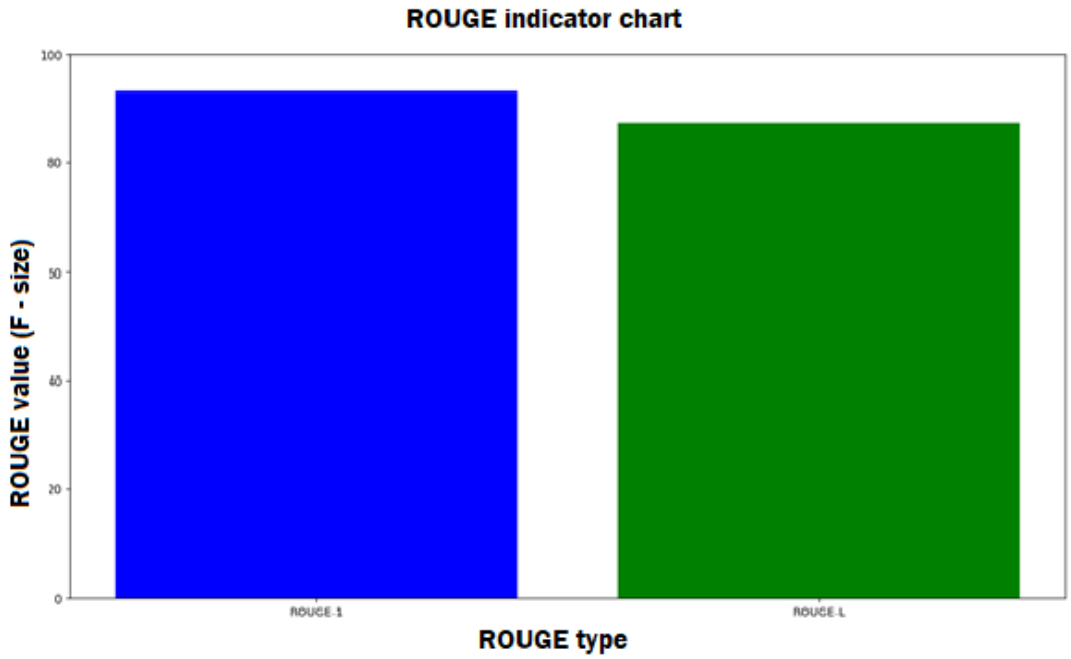

Figure 3.
Histogram of ROUGE readings.

The high values of these two indicators, shown in the histogram, indicate that the model worked effectively to preserve the structure and content of the text, as well as the high quality of the processed texts.

The evaluation of the model is not limited to evaluating its performance, but also determines the steps necessary for its improvement. For example, based on the data obtained from the test results of the model, there may be a need to add additional algorithms or expand the data to improve its functions. To improve the model, it will often be necessary to retrain, supplement with new data, or change hyperparameters. This process – opens the way for the continuous development of the system.

The efficiency and quality of each stage affect the overall performance of the system. The training, testing, and evaluation processes are each important and necessary. Only with the correct implementation of these processes will the work done make the system more accurate and efficient. For example, in systems that work with large amounts of data using neural networks, the location and role of each stage are unique.

The performance of the model will also depend on its computational capacity, time savings and efficient use of resources. In this sense, the training and testing times of the system, as well as the hardware resources necessary for them, also play an important role. For example, when working with large data sets, the training time can take several hours or even days. For the same reason, the performance and efficiency of the system is assessed not only by the accuracy of the model, but also by the speed of its operation and the efficiency of resource use.

Various techniques can be used when testing and evaluating the performance of the model. These include, for example, checking response time, resource consumption, and system scalability. Especially for systems that work with large amounts of data, these parameters are very important, as they determine the availability and speed of operation of the system.

The importance of test data is especially significant when evaluating the model. Test data is required, particularly for model testing with previously unfamiliar data. This data should be highly correlated with real life, as only in this way can it be determined how the model will perform in real tasks in the future. The evaluation results of the model demonstrate how compatible the data is with real life. By increasing the quality and quantity of test data, the capabilities of the system can be further enhanced.

The efficiency of the main components of the system increases not only the expenditure of time and resources, but also the quality of the product and the possibilities of its application in real life. Therefore, the stages of training, testing and evaluation complement each other, ensuring the correct and effective functioning of the system.

Evaluation of the quality of annotations

For an objective assessment of the quality of the created annotations, a comparative examination was conducted based on the generally accepted metrics of automatic summarization and machine translation: ROUGE, BLEU, and METEOR.

These metrics allow you to quantify the accuracy, completeness, and semantic consistency of automatically generated annotations compared to reference (manual) annotations.

Experimental setup

A corpus of 100 scientific articles covering the fields of biomedicine, computer science, and engineering was used as a test set. A reference abstract was prepared manually for each article, and then an automatic annotation was generated using the following models:

- Model 1 (our system): Cohere API + Gensim + NLTK (Hybrid approach).
- Model 2: BERTSUM (Transformer model, fine-tuned).
- Model 3: TF-IDF + LSA (Classical extractive approach).
- Model 4: OpenAI GPT-3.5-turbo (Zero-shot summarization).
- Model 5: Manual annotation (Used as a reference).

Table 2.
Metrics and results.

Model	ROUGE-1	ROUGE-L	BLEU	METEOR
Our system (Cohere+Gensim)	0.52	0.47	0.41	0.39
BERTSUM	0.49	0.44	0.37	0.35
TF-IDF + LSA	0.33	0.28	0.24	0.20
GPT-3.5-turbo	0.50	0.46	0.40	0.37
Manual annotation (standard)	1.00	1.00	1.00	1.00

The data obtained show that the proposed hybrid system, based on a combination of classical text processing algorithms (Gensim, NLTK) and modern transformers (Cohere API), demonstrates the highest performance among automatic systems. The increase in the BLEU and METEOR metrics is particularly noticeable, which indicates a high lexical and semantic consistency with the reference annotations.

The BERTSUM model, despite its architectural complexity, is inferior to our system in terms of accuracy, which may be due to its lower adaptation to multilingual buildings and the lack of additional heuristics. Classical methods have shown the lowest performance. The results of GPT-3.5 are close to our system, but without taking into account the structure of scientific texts.

The results of the experimental evaluation confirm the high efficiency of the developed automatic annotation system, which combines the advantages of both statistical and neural network approaches. The obtained metrics indicate the prospects of a hybrid architecture for the tasks of summarizing scientific texts. In the future, it is planned to further adapt the system to various scientific domains (biology, medicine, humanities) and expand language support.

4. Discussion

The proposed hybrid system for automatic annotation of scientific texts demonstrates high results due to a combination of traditional text processing algorithms (Gensim, NLTK) and modern transformers (Cohere API). Experimental indicators ROUGE-1 (0.52), BLEU (0.41), and METEOR (0.39) confirm the superiority of the system over classical methods such as TF-IDF and LSA, as well as over specialized transformer architectures such as BERTSUM. A significant advantage lies in multilingual adaptability and the ability to correctly process both English-language texts and texts in other languages. The hybrid structure also allows for maintaining a high degree of semantic consistency of annotations and better handling of the context than isolated models.

In addition, the architecture of the system provides flexibility and scalability, which makes it suitable for use in real research projects. The implementation of quality control, data storage, and secure access modules increases trust in the system and allows for efficient management of large amounts of scientific information. Special attention is paid to the user interface and data security, including support for authorization, encryption, and separation of rights. In the future, it is planned to expand language support, adapt to various scientific fields (for example, biomedicine and humanities), and further improve algorithms in the direction of increasing interpretability and transparency of models.

5. Conclusion

The development of an automated annotation system for scientific texts represents a significant step forward in the field of natural language processing. The proposed hybrid model successfully integrates the strengths of both classical statistical approaches and modern neural network algorithms. Thanks to the use of the Cohere API, Gensim, and NLTK libraries, the system demonstrates high accuracy and semantic relevance, which is confirmed by the results of the ROUGE, BLEU, and METEOR metrics. High accuracy in determining the language and adequate transmission of the meaning of the source texts makes the system an effective tool for scientific summarization.

In addition to high-quality annotations, the system offers extensive functionality, from a user-friendly interface to scalability and security modules. The ability to integrate with other platforms and work with multilingual corpora makes it a universal solution for academic and research environments. In the future, it is planned to develop the project in the direction of cloud infrastructure, expansion of language support, and adaptation to various scientific disciplines, which will significantly speed up and simplify work with scientific information in the era of its exponential growth.

References

[1] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345*, 2019. https://doi.org/10.48550/arXiv.1908.08345

[2] A. Cohan, W. Ammar, M. Van Zuylen, and F. Cady, "Structural scaffolds for citation intent classification in scientific publications," *arXiv preprint arXiv:1904.01608*, 2019. https://doi.org/10.48550/arXiv.1904.01608

[3] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," presented at the International conference on machine learning, 2020.

[4] J. Lee *et al.*, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234-1240, 2020. https://doi.org/10.1093/bioinformatics/btz682

[5] Y. Gu *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1-23, 2021. https://doi.org/10.1145/3458754

[6] M. Yasunaga, Y. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and J. Leskovec, "ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks," *arXiv:1910.05786). arXiv*, 2019.

[7] S. Subramanian, R. Li, J. Pilault, and C. Pal, "On extractive and abstractive neural document summarization with transformer language models," *arXiv preprint arXiv:1909.03186*, 2019. https://doi.org/10.48550/arXiv.1909.03186

[8] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017. https://doi.org/10.48550/arXiv.1705.04304

[9] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," presented at the Proceedings of the AAAI Conference on Artificial Intelligence, 2017.

[10] A. Gidiotis and G. Tsoumakas, "A divide-and-conquer approach for neural scientific document summarization. In J. M. Jose et al. (Eds.)," in *Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020). Springer*, 2020, pp. 448–463.

[11] O. O. Amusat *et al.*, "Automated annotation of scientific texts for ML-based keyphrase extraction and validation," *Database*, vol. 2024, p. baae093, 2024. https://doi.org/10.1093/database/baae093

[12] L. Beasley and P. Manda, "Comparison of natural language processing tools for automatic gene ontology annotation of scientific literature," (No. e27028v1). PeerJ Preprints, 2167-9843, vol. 6 2018.

[13] K. M. Khelif, R. Dieng-Kuntz, and P. Barbry, "Mining biomedical texts to generate semantic annotations," Doctoral Dissertation, INRIA, 2007.

[14] S. Serikbayeva *et al.*, "Development of a model and technology of access to documents in scientific and educational activities," *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 2–114, pp. 44–58, 2021.