Probabilistic thematic modelling of Ukrainian-language texts based on the Latent Dirichlet Allocation algorithm

Victoria Vysotska[†], Denys Ptushkin[*,†], Rostyslav Fedchuk[†] and Roman Lynnyk[†]

Lviv Polytechnic National University, Stepan Bandera 12, 79013 Lviv, Ukraine

Abstract

The article presents the results of the study of methods of thematic modelling of texts using the Latent Dirichlet Allocation (LDA) algorithm for the Ukrainian-language corpus of documents. The proposed model allows you to automatically detect hidden topics in large volumes of unstructured text data without prior labelling. The model was implemented in Python using Gensim and pyLDAvis libraries. Perplexity and coherence metrics were used to assess the quality of the model, which showed that the optimal number of topics depends on the characteristics of the corpus and the parameters of hyperparameters α and β. Texts and demonstrate the suitability of the method for a wide range of applied tasks – analysis of user reviews, media analytics, classification of scientific publications and monitoring of social networks. A comparative study with alternative approaches (K-means, NMF, BERTopic, transformer models) was carried out, which showed that LDA provides the best balance between interpretation, speed and computational efficiency. The developed program module "Thematic Analysis Module" implements an automated system for thematic modelling, which can be used both in scientific research and in analytical information systems.

Keywords

thematic modelling, Latent Dirichlet Allocation, LDA, natural language processing, machine learning, probabilistic model, coherence, TF-IDF, Gensim, Ukrainian-language corpus of texts

1. Introduction

Today, humanity is in an information glut: a massive amount of text data is generated every day – news, scientific publications, messages in social networks, forums, blogs, and instant messengers. This information is often unstructured and challenging to subject to classical analysis, which causes the need for automated tools to classify, sort, filter, and understand it. One of the most effective modern methods of analysing such texts is thematic modelling. It allows you to automatically detect topics hidden in texts based on the probabilistic distribution of words. For example, without having to read thousands of product reviews, you can automatically discover that people most often talk about "price", "quality", "delivery", "packaging", etc. This approach is actively used in the following areas:

- Journalism and media analytics – to track information campaigns, trends in the media;
- Business and marketing – to analyse user reviews, surveys, customer feedback;
- Science – classification of scientific publications by topics;
- Public administration – monitoring of public moods, thematic appeals of citizens;
- Education – automatic classification of educational materials.

[*]*AISSLE-2025: The International Workshop on Applied Intelligent Security Systems in Law Enforcement, October, 30 – 31, 2025, Vinnytsia, Ukraine*

[1,*] Corresponding author.

[†] These authors contributed equally.

Victoria.A.Vysotska@lpnu.ua (V. Vysotska); denys.ptushkin.sa.2022@lpnu.ua (D. Ptushkin); rostyslav.b.fedchuk@lpnu.ua (R. Fedchuk); roman.o.lynnyk@lpnu.ua (R. Lynnyk)

0000-0001-6417-3689 (V. Vysotska); 0000-0003-0374-8359 (D. Ptushkin); 0009-0002-6669-0369 (R. Fedchuk); 0009-0007-0948-4338 (R. Lynnyk)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2025-12-11

CEUR-WS.org/Vol-4126/paper16.pdf

- Thus, thematic modelling is one of the key natural language processing (NLP) tools, allowing you to efficiently work with large text arrays without the need for manual processing.

The purpose of this work is the in-depth development of information technology for thematic modelling of texts, as well as the practical implementation of the thematic model on a specific corpus of Ukrainian-language documents using Python tools. During the work, it is planned to investigate how the pre-processing of texts, the choice of the number of topics, algorithms and parameters affects the quality of the thematic model, as well as to analyse the practical results of modelling and the possibilities of their application in a real environment.

To achieve the goal, it is necessary to solve the following tasks:

- Analyse the literature on thematic modelling (LDA, NMF, PLSA).
- Choose a corpus of texts for modelling (e.g., news, articles, forum posts).
- Clean up the data – remove HTML tags, numbers, punctuation, stop words.
- Perform lemmatisation or stemming (if necessary, in Ukrainian).
- Create a Bag-of-Words or TF-IDF matrix.
- Build an LDA model with a different number of themes.
- Visualise the results obtained.
- Analyse the interpretation of topics.
- Compare the quality of models by coherence.

The object of research is the text corpus – a set of documents containing natural language (in our case, Ukrainian). These can be news, social messages, reviews, scientific articles, product descriptions, etc. Such texts are unstructured, which makes it difficult to analyse them without pre-processing. That is why the object of research is interesting from the point of view of practical data processing. The subject of the study is algorithms and methods of thematic modelling, in particular:

- Latent Dirichlet Allocation (LDA);
- Non-Negative Matrix Factorisation (NMF);
- Probabilistic Latent Semantic Analysis (PLSA);
- TF-IDF and Bag-of-Words for text representation;
- Quality assessment metrics: coherence, perplexity.

Although thematic modelling is a well-known technique, its application to Ukrainian-language texts has not yet been sufficiently researched. Most libraries and examples focus on English-language content. Therefore, the novelty of this work lies in:

- Implementation of thematic modelling specifically for the Ukrainian language;
- Comparison of models with a different number of themes for a real case;
- Application of modern methods of pre-processing of Ukrainian-language texts (for example, through langdetect, pymorphy2-uk or Stanza);
- visualisation of results and analysis of the correspondence of topics to the real content of documents.

Also, the novelty lies in the application of coherence to automatically assess the quality of the model without human intervention. The developed model has a number of real-world applications:

- Information systems (filtering news, searching by topics, classification of documents).
- Education (automatic grouping of educational materials by topic).

- Marketing (classification of customer reviews on topics to identify pain points).
- Science (analysis of scientific publications and identification of new research trends).
- Security (monitoring social media to identify radical topics).
- Electronic democracy (analysis of citizens' appeals in petitions, complaints, forums).

The model is universal and can be adapted to any subject area containing large amounts of textual information. It is necessary to investigate the methods of thematic modelling of texts, in particular the Latent Dirichlet Allocation (LDA) algorithm, which allows you to automatically identify the main topics in a large amount of text data. The preliminary processing of the corpus of documents was carried out, a thematic model was built, and its results were analysed. The study confirmed the effectiveness of thematic modelling as a tool for classifying and analysing unstructured texts. The practical implementation of the model has demonstrated that this approach can be used in various fields – from journalism and marketing to science and education. The results obtained showed the dependence of the quality of thematic modelling on the preliminary processing of data, the choice of the number of topics and the parameters of the model. Thus, the work contributed to the consolidation of knowledge in computational linguistics and the acquired practical skills in natural language processing.

2. Related works

In today's information age, society generates enormous amounts of text data every day. News sites, social networks, forums, emails, blogs, user reviews, documents – all this creates a powerful flow of information that needs to be stored, processed and analysed. According to analytical agencies, tens of millions of new texts of various formats are created every day in the world, and this trend is only growing, requiring manual analysis. There is an urgent need for tools that can automatically reveal meaning and structure in unstructured text.

One of the most promising areas in this area is thematic modelling of texts – a method of identifying hidden thematic structures in large amounts of text data. Thematic modelling allows you to understand what the documents are about, without the need to read them thoroughly. It automatically classifies texts by content, highlights key topics, and allows you to visualise the results, which significantly simplifies analysis.

The principle of thematic modelling is that each document consists of a particular set of topics, and each topic consists of a specific set of words. For example, if the system analyses the news corpus, it can detect issues such as "politics", "economy", "sports", "education", even if these labels are not set manually. Thematic modelling algorithms, in particular Latent Dirichlet Allocation (LDA), are based on statistical patterns of the joint appearance of words in texts and are able to automatically find relationships between words and group them into meaningful topics. The relevance of this topic is due not only to the rapid growth of textual data but also to the need to interpret it effectively. In many fields, from media and journalism to education, marketing, and research, thematic modelling is becoming an indispensable tool. It allows:

1. Analyse large amounts of news;
2. Identify trends in social networks;
3. Carry out automatic classification of documents;
4. Segment customer reviews by topic;
5. Build dashboards for decision-making.

Latent Dirichlet Allocation (LDA) is a classical probabilistic generative model of the issues proposed in [1]. LDA formalises a document as a mixture of topics and a topic as a distribution of words; It was this work that laid the mathematical foundation for most of the further research in

case study. Its advantages are ease of interpretation, relative ease of implementation, and low hardware requirements; The disadvantage is the weak ability to capture context (sequence/order) and problems with short texts.

Other classical methods – PLSA and NMF – use linear/probabilistic factorisations of the document-term matrix [2]. NMF sometimes gives more stable and interpreted themes on a small corpus, but lacks Bayesian regularisation of LDA and can be sensitive to noise. Comparative studies show that no "classic" dominates universally – the choice depends on the size of the case, the length of the documents and the goals of the analysis.

Modern approaches: built-in representations and hybrids:

1. BERTopic is a practical cluster-embedding approach that combines transformer embedding of documents (BERT-like) with dense clustering and c-TF-IDF for describing topics [3]. BERTopic shows good semantic coherence of issues, especially on short and variable texts (tweets, comments), but requires more resources and depends on the quality of embeddings.

2. Contextualised Topic Models (CTM) and their development – methods that combine the BOW part with contextual embedding (BERT) into variational autoencoders [4]. They increase the coherence of topics compared to classical LDA, especially on data where context significantly changes the meaning of words. CTM and derivatives (improvement due to negative sampling, pre-training, etc.) are now actively researched and often give better NPMI/UMass results than LDA.

3. Top2Vec / embedding-based clustering – an approach where documentary and verbal embedding are used to simultaneously identify topics and semantic centres (without an explicit K task). It works well for large enclosures with moderate document lengths. The downside is that interpreting topics sometimes requires additional c-TF-IDF or manual filtering.

The general trend in recent years has been to replace or supplement purely frequency representations (BoW/TF-IDF) with contextual embeddings (BERT, MiniLM, etc.). It improves the quality of topics (semantics), but increases computational costs and can complicate interpretation in some cases [5].

In comparative work, a set of metrics is most often used: perplexy (probabilistic measure), coherence (UMass, Cv, C_v) and NPMI [2]. Almost all modern research emphasises that perplexy and coherence sometimes conflict (perplexy can decrease, while coherence can deteriorate), so it is recommended to use a combination of metrics to choose the optimal model and number of topics.

Although most of the methodological works are tested on English-language corpora (20 Newsgroups, Wikipedia, ArXiv abstracts), there are more and more publications dedicated to Ukrainian-language corpora. Study of themes in folk songs of Podillia (case-study) [6] – LDA applied to folklore texts; The authors note the importance of lemmatisation and morphological normalisation through productive word formation in the Ukrainian language [6]. An analysis of discussions and media coverage of the war (Russo-Ukraine war) showed [7] that for social networks/tweets, it is advisable to compare LDA with models on embedding (BERTopic/CTM): transformer approaches are better at catching context and nuances, while LDA gives more stable "dark" clusters for a large number of short, noisy messages. These studies emphasize two important theses for Ukrainian: (1) pre-processing (lemmatization, removal of inflectional forms, stop words) significantly affects the quality of topics; (2) the choice of model depends on the genre of texts – for long forms (articles), LDA/NMF work well; for short/social media – CTM/BERTOPIC/Top2Vec gives a better semantic grouping [7].

Table 1
Generalised comparison [1-8]

Method	Main idea	Strengths	Weaknesses	Recommendations for use (Ukrainian)
LDA	Bayesian generative model	Interpretation, low resources	Weak context, problems with	Basic method-reference; suitable for long

	(BoW)		short texts	documents; requires lemmatisation.
NMF	Linear factorisation of document–terms	Simplicity, sometimes better stability with small data	Sensitivity to noise, no a priori	Alternative to LDA on small cases
BERTopic	Embedings + Clustering + c-TF-IDF	High coherence, better for short texts	Needs embeddings (resources), fewer "clean" topics	Social networks, comments; use with Ukrainian embedings (mBERT, uk-BERT if available) [3]
CTM / context-ualised	BOW + BERT embedding (VAE)	Context-sensitive, better NPMI/ coherence	More complex implementation , resource intensity	When semantics/discourse is essential, it gives strong results on multilingual corpora. [8]
Top2Vec / embedding clustering	Coordination of documents and words in embedded space	Does not need a previous K, good semantic centres	Interpretation is sometimes more complicated	Large body, quick overview of themes

LDA remains a "practical standard" – it provides interpreted topics and serves as a good baseline for any thematic analysis [1]. It is especially valuable because of the limited resources or the need to explain the results to a non-professional audience.

Contextual models (CTM, BERTopic) show a marked improvement in the clinical (semantic) quality of topics (NPMI/Cv), especially on short or highly contextual texts [3]. If the project allows for calculation costs, these approaches give a better interpretation of the topics.

Assessment should be multidimensional [2]. Perplexy ≠ coherence: in practice, it is advised to minimise perplexy and simultaneously maximise NPMI/Cv (or perform human validation for the most important topics).

The peculiarities of the Ukrainian language (morphology, inflexion, word formation) make high-quality linguistic preprocessing critical: tokenisation, lemmatisation (pymorphy2-uk / Stanza / spaCy pipelines), removal of stop words, and filtering of N-grams. Studies on Ukrainian corpora confirm that without such training, the quality of topics drops sharply [6].

Practical recommendations for research:

1. Implementation of LDA as a baseline (Gensim) after careful linguistic pre-processing (lemmatisation, stop words, removal of frequent noise tokens), estimation of perplexity and NPMI/Cv on the K grid [1].

2. BERTopic testing (with Ukrainian/multilingual embeddings – mBERT or lightweight MiniLM models) and CTM – comparison of NPMI and Cv; on short texts, BERTopic is expected to win [3].

3. Validation: In parallel, a small manual assessment (human judgment) for 10-20 topics will give a high-quality check of metrics.

4. Resources: if there are resource constraints – use LDA/NMF; Transformers launch – CTM/BERTOPIC will give better semantics.

5. Documentation: fixing hyperparameters (α, β, minimum frequency of terms, seed) so that the results are reproducible [9].

Classical approaches to topic modelling (LDA, PLSA, NMF) formalise a document as a mixture of topics and a topic as a distribution of words, and they are widely used as a baseline in thematic analysis studies. Having settled on LDA, we are guided by its interpretation and stability in problems with significant cases. Modern approaches (BERTopic, Contextualised Topic Models, Top2Vec) combine contextual embedding and clustering, which allows you to increase the semantic coherence of topics, but requires more computing resources. Evaluation of models is carried out by perplexity and coherence (UMass, NPMI, Cv), since the combined approach to validation gives the most reliable results. For Ukrainian-language corpora, the importance of lemmatisation and morphological normalisation is additionally emphasised." (case studies: Blei et al. 2003; BERTopic; CTM and comparative studies).

Within the framework of this work, the development of a thematic model of texts is considered, which allows you to automatically single out key topics from a large corpus of Ukrainian-language texts. The focus is on the LDA (Latent Dirichlet Allocation) algorithm, which is one of the most common and at the same time interpreted methods of thematic analysis. A comparison of this approach with other methods such as clustering, classification and modern neural approaches (transformers) will also be carried out, and the advantages and disadvantages of each technique will be identified. Special attention is paid to the formulation of the problem that the proposed thematic model is designed to solve. First of all, it is about automating the understanding of text data in situations where labels are missing, and human analysis is too costly or impossible. Thus, this section lays the theoretical and methodological basis for the implementation of the work, demonstrating not only technical aspects but also the strategic significance of thematic modelling in the digital information age.

Within the framework of this work, a tool for thematic modelling of texts is being developed, the primary purpose of which is to automatically identify content topics in the corpus of documents without preliminary markup or manually specified categories. This approach allows you to better understand the structure and content of large text arrays, identify hidden patterns, and optimise the content analysis process.

The product being developed is a thematic model built using the Latent Dirichlet Allocation (LDA) algorithm, which belongs to the category of probabilistic models. LDA allows each document to be represented as a combination of several topics, and each topic as a set of keywords with appropriate weights. Based on the statistics of the coincidence of words in different documents, the model identifies those words that occur most often together and groups them into topics. This approach is beneficial in cases where the structure of the texts is not strictly defined, and manual classification is too costly or subjective. So that they cover a wide range of topics, including politics, technology, education, health, economics, etc, this choice is justified by the fact that in real conditions, the texts are of a mixed nature and often include several topics at the same time, so high-quality thematic modelling should take this context into account. Preliminarily, texts undergo standard processing: clearing punctuation and special characters; lowercase casting; removing stop words; lemmatisation (if necessary); tokenization. The product is developed in the Python programming language, using the following libraries:

1. Gensim – a library for building LDA models and working with text data;
2. pyLDAvis – visualisation of the constructed theme (interactive graphs, which show the placement of topics in vector space);
3. NLTK / spaCy / Stanza – for pre-processing of texts: tokenisation, lemmatisation, removal of stop words;
4. Pandas – convenient work with text datasets;
5. Matplotlib / Seaborn – Additional visualisation of results is needed.

This stack of tools allows you to effectively implement a complete cycle of thematic modelling – from word processing to visual analysis of results.

In the field of natural language processing (NLP), there are several methods that, to some extent, perform the function of grouping, classifying or summarising text documents. Although thematic modelling, in particular based on LDA, is a specialised approach to identifying topics, it is worth considering other methods that can act as its counterparts in specific contexts.

1. K-means clustering is one of the most popular methods of unsupervised learning, which distributes objects (in our case, documents) into groups called clusters. The algorithm tries to minimise the distance between documents within the same cluster and maximise the distance between different clusters. Each document is represented as a vector (for example, based on TF-IDF), and the cluster itself is defined through the centre of mass. Advantages:

- Easy to implement and quick learning.
- Does not require labels (unsupervised).
- Scales well for large amounts of text.

Disadvantages:

- Clusters do not have a clear, meaningful description (there is no list of words as in LDA).
- It is challenging to interpret what each cluster is about.
- Does not take into account topics, only "groups of similar texts".

K-means groups texts by similarity, while LDA detects semantic themes within texts. Clusters are "similar documents", topics are "similar words".

2. Classification of texts (SVM, Naive Bayes) – these algorithms belong to supervised learning, which requires pre-labelled data. Each text must have a predefined category (e.g. "sports", "education", "politics"), and the model learns to recognise these categories with new examples.

Advantages:

- High accuracy with proper data preparation.
- Easy to use (especially Naive Bayes).
- Works well with short texts.

Disadvantages:

- Does not work without labels – you need to manually classify a large number of documents for training.
- It does not detect new topics; it works only with those that are already known.
- Less flexible in a dynamic environment (changing topics requires retraining).

The classification requires tagged training data, while LDA is fully automated and suitable for exploring new, previously undefined topics.

3. Transformers (BERT, GPT, BERTopic) – transformer-based models are modern approaches in NLP that allow you to take into account the context of an entire sentence or text. Models like BERT (Bidirectional Encoder Representations from Transformers) generate vector representations of texts that preserve semantics at a deeper level. BERTopic is an example of thematic modelling that combines BERT and clustering. Advantages:

- High-quality results.
- Taking into account the context and order of words.
- The ability to analyse the nuances of language, synonyms, and irony.

Disadvantages:

- Need for powerful hardware (GPU/TPU).
- Complexity of implementation (not "out of the box").
- Weak interpretation (results are difficult to explain – "black box").

Transformers are more potent in quality, but more challenging to implement. LDA loses in accuracy, but wins in simplicity, interpretation, and resources.

4. Alternative thematic models of NMF and PLSA. NMF (Non-negative Matrix Factorisation) decomposes the document-term matrix into two smaller matrices that reflect topics and word distribution. It works similarly to LDA, but is based on linear algebra rather than probabilities. Advantages:

- A simple approach without complicated statistics.
- Can give clear topics for minor cases.

Disadvantages:

- Themes are less stable when data changes.
- Less interpreted compared to LDA.

PLSA (Probabilistic Latent Semantic Analysis) is a precursor to LDA. A statistical model that also identifies topics by word distribution in documents. Advantages – considered the "foundation" for LDA – are theoretically powerful. Disadvantages:

- The model is prone to overtraining.
- Doesn't scale to large amounts of data.
- Does not allow you to simulate new documents without re-learning.

NMF is simpler, PLSA is theoretically deeper, but both are inferior to LDA in flexibility, scalability, and resilience.

Table 2
Comparison of approaches

Approach	Principle	Advantages	Disadvantages
Rule-based	Hard-coded rules	Accuracy in narrow tasks	Not scalable
SVM Classification, Naive Bayes	Labels needed	Works well with labels	Inability to work with unstructured topics
Clustering (K-means)	Unattended	Simplicity	Themes are often uninterpreted

Table 3
Feature analysis

Sign	Thematic Modelling (LDA)	Transformers
Interpretation	High	Low (black box)

Speed	High	Slow on large cases
Quality	Average	High, but needs fine-tuning
Resources Required	Small	High (GPU, TPU)
Explanation of results	Themes can be interpreted	Difficult to explain the reason for the classification

Thematic modelling, especially in the implementation of Latent Dirichlet Allocation (LDA), has a number of significant advantages that make it a versatile and convenient tool for analysing large corpora of text data. Unlike other approaches (e.g., classification or transformers), LDA strikes an optimal balance between interpretation, automaticity, and efficiency.

1. Unsupervised learning. One of the most valuable properties of thematic modelling is its independence from the labelled data. The algorithm does not require prior manual classification of documents – that is, there is no need to create a training sample, where each text is manually assigned to a specific topic. It is essential in cases where:

- Labels are difficult or expensive to obtain.
- The subject matter of the data changes over time.
- It is necessary to explore a new, unexplored corpus of texts.

Thus, thematic modelling is an indispensable tool for exploratory analysis, when it is necessary to find out: "what the texts are about", and not just classify them into already known categories.

2. Visualisation capability. Modern libraries, including pyLDAvis, make it easy to visualise the results of thematic modelling. It opens up opportunities for intuitive analysis even for users without technical training. Thanks to visualisation, you can:

- See how topics are placed in a vector space.
- Evaluate which words are key for each topic.
- Check which documents belong to which topics and how strongly.
- Explore the intersections between topics (the more topics overlap, the more similar they are).

It makes thematic modelling a powerful tool for data analytics and presentation.

3. Flexibility of the model. The user independently sets the number of topics that the model should find. It allows you to adapt the analysis to different tasks:

- If you need to conduct a general review, you can choose a smaller number of topics (for example, 5-10).
- If you need details, the model can be reconfigured for 20-30 themes.
- In addition to the number of themes, you can flexibly customise:
- the number of keywords in the topic;
- alpha and beta distribution parameters (affecting the "smearing" of topics across documents);
- filtering of rarely used or commonplace words.

This flexibility allows you to optimise the model for a specific type of content or business task.

4. Interpretation of results. Unlike many modern models (especially transformers), LDA provides transparency in the results. Each topic is clearly expressed in the form of a set of words,

and each document has a distribution of issues with the weight of each of them. It makes it possible to:

- Quickly describe the essence of the topic (by keywords).
- Understand how the content of the document is related to the issues.
- Check the logic of the results based on human intuition.
- Justify the conclusions of analytics to customers or management.

LDA models are one of the few in machine learning that can be explained and defended in front of a non-professional audience.

5. Efficiency and the possibility of the use of small resources. LDA models do not require significant computing power. They can be launched:

- on a regular laptop or server without a GPU;
- with small corpora of texts (even several hundred documents);
- with limited RAM.

It opens up access to thematic analysis for small companies, research projects, and university laboratories. Even for educational purposes, LDA is an excellent demonstration of how text analytics works in the real world.

In addition to the implementation of thematic modelling through open libraries (for example, Gensim and LDA), there are many ready-made commercial or SaaS solutions on the market that provide the functionality of automatic analysis of text topics. Such services, as a rule, are aimed at business intelligence, automation of feedback processing, customer requests, social networks, etc. Below is a detailed review and comparison of the most well-known platforms.

IBM Watson Natural Language Understanding is a platform with a set of tools for natural language processing. One of its components, topic classification, allows you to identify common topics in the text (for example, politics, healthcare, finance). In this case, the thematic analysis is based on pre-trained classifiers. Advantages:

- Support for many languages.
- High-quality results.
- The API integrates seamlessly into business systems.
- It provides not only themes but also emotional tone, categories, concepts, and objects.

Disadvantages:

- Only works with a fixed list of topics.
- There is no full-fledged topic-forming model (as in LDA).
- Commercial model: paid for a large number of requests.
- Limited flexibility for the user (no access to simulation engines).

Google Cloud Natural Language API – Google's word processing service includes content classification, where documents are classified according to a hierarchy of ~700 topics (for example, /Arts & Entertainment/Music or /Business/Banking). It is based on deep neural networks and a predefined topic dictionary. Advantages:

- Reliability and speed from Google.
- An extensive database of topics and subtopics.
- Convenient to integrate into cloud services.

- Support for many formats.

Disadvantages:

- Topics are hardcoded – it is impossible to identify new ones.
- Interpretation of the result is only possible within the Google framework.
- There is no transparency – it is not clear which words influenced the classification.
- The cost increases when processing large arrays of texts.

MonkeyLearn is a cloud-based platform for text analysis that allows you to create your own classifiers and pre-trained thematic templates. It is positioned as a no-code/low-code tool for business users. Advantages:

- Ready-made templates (for example: customer support, surveys, e-commerce).
- You can create custom models without programming.
- Visual interface for customising categories.
- It has an API and integration with Google Sheets, Zapier, etc.

Disadvantages:

- The free version is minimal.
- Less flexible for complex analysis.
- It is not a full-fledged thematic modelling (works as a classifier).

Gensim (LDA implementation) is an open-source Python library for thematic modelling. Implements LDA (Latent Dirichlet Allocation), as well as other methods for analysing the latent structure of texts. Supports model training both in memory and from streaming data. Advantages:

- Complete freedom of customisation: number of topics, words, alpha/beta.
- Open core – can be expanded and adapted.
- Visualisation capability (via pyLDAvis).
- Works locally, without cloud costs.

Disadvantages:

- Requires programming (not suitable for non-specialists).
- Requires independent processing of texts (cleaning, tokenisation, etc.).
- It does not have a graphical interface "out of the box".

BERTopic is a modern library for thematic modelling that combines contextual vector representations of BERT and clustering (e.g. HDBSCAN) to detect topics. Topics are created based on the similarity of vectors of texts. Advantages:

- It takes into account the context that "bank" and "riverbank" will not be on the same topic.
- A more accurate model for short or unstructured data.
- Topics can have dynamic depth (topics within topics).
- It has integration with visualisation and meta-information.

Disadvantages:

- Requires a lot of resources (GPU for fast work).
- Complexity of installation (need for transformers, BERT models).
- The interpretation is more complicated than that of the classic LDA.

3. Problem formulation

In the XXI century, humanity lives in the conditions of the information revolution: text data is created at an unprecedented speed – in news, social networks, instant messengers, reviews, reports, blogs, comments, documents. All this forms a complex and extensive information ecosystem that requires tools for systematisation, analysis, and understanding. Much of this information is unstructured – i.e., one that does not have clear tags, headings, or categories- and therefore, it is difficult to process automatically using traditional methods.

Modern society faces the challenge of efficiently processing large amounts of unstructured text data. Classical methods of analysis – for example, manual classification, keyword search, rule-based systems – are not able to scale to large data sets and do not allow you to automatically identify content topics without prior human intervention. It makes it difficult:

- Decision-making in business, science, journalism;
- Identifying trends and topics in social networks;
- Customer feedback analytics;
- Creation of personalised recommendation systems.

Classic classifiers (SVM, Naive Bayes) require labelled data, which means that someone has to manually specify which topic each document belongs to. In cases where thousands of texts are used, it becomes an irrelevant, expensive, and slow process. In turn, clustering algorithms (for example, K-means), although they allow you to group documents, do not provide interpreted topics – we cannot say "what" each cluster is about without additional analysis. In connection with the described problem, the product under development faces several tasks:

1. Develop a thematic model that automatically highlights topics from the corpus of documents. The model must work without previous labels, and therefore belongs to the unsupervised learning class. The algorithm should determine the most likely topics in a large corpus based on statistical patterns of word distribution.

2. Ensure the interpretation of the results. Unlike the "black boxes" of modern deep learning, the developed system should provide clear and transparent results. A list of keywords should express the topic, and each document should show which topics are present in it and in what ratio.

3. Provide a flexible tool that works even without labelled data. The thematic model should work on any corpus of texts – news, forums – without the need for mark-up. It should be scalable, customizable (for example, change the number of themes) and available for local launch, without cloud dependency.

4. Compare several approaches and justify the feasibility of choosing LDA. In order for the choice of thematic modelling (in particular, the LDA algorithm) to be reasonable, it is necessary:

- Compare it with classifiers, clusters, and transformers.
- Assess the advantages and limitations of other approaches.
- Show that LDA is the best compromise between interpretation, automation, and technical simplicity.

A comprehensive study of thematic modelling of texts as a modern tool for analysing large volumes of unstructured data has been carried out. The main goal was to create a model capable of automatically detecting content topics in the body of documents without pre-labelling, as well as comparing this approach with similar methods. A thematic model based on the Latent Dirichlet Allocation (LDA) algorithm using Gensim and pyLDAvis libraries has been developed. The corpus

of texts used has undergone a complete cycle of pre-processing: tokenisation, cleaning, deletion of stop words, and, if necessary, lemmatisation. After building the model, a set of topics was obtained, each of which is described by a list of words with the highest probability, and an analysis of the distribution of topics across documents was carried out.

4. Methods

Topic modelling as a task of extracting hidden topics in the corpus of texts has become one of the key paradigms in natural language processing and text data analysis. This section provides an overview of the three main approaches – classical generative models, factorisation-based models, and modern approaches with contextual embedding – with a focus on their applicability, strengths and weaknesses, and challenges for Ukrainian-language corpora.

The original and most common approach is the Latent Dirichlet Allocation (LDA), proposed in [1]. The model formalises the document as a mixture of topics, and the topic as a distribution of words, using a priori Dirichlet distributions for θ (document-topic) and φ (topic-words) among the advantages: high interpretation of themes, support for a large corpus, and relatively simple implementation. However, a number of studies have noted the weak ability of the model to take into account word order, context or short texts, as well as the instability of the results regarding initialisation or order of documents [10]. Other researchers in their review classify LDA as the "dominant" model in the topic of topic modelling before the era of deep learning [11]. The problem of the "order-effect" in LDA was also investigated, and the LDADE approach for adjusting hyperparameters to reduce the instability of topic distributions was proposed [10].

Along with LDA, methods based on negative matrix factorisation (NMF), latent-semantic analysis (LSA), and pLSA are widely discussed in the literature. These methods, although not as common as LDA, sometimes show better stability in small enclosures or with limited resources. A study [12] compared LDA, NMF, and embedding clustering on tweet data and found that traditional models have limitations and are less stable in short texts. Some works also pay attention to dynamic versions of topic models (e.g. Dynamic Topic Model, HDP) to track thematic changes over time [13].

In recent years, models that combine embedding text representations (e.g., BERT-like models) with clustering algorithms or variational autoencoders have been growing in popularity. For example, BERTopic uses document embedding (based on transformers), then UMAP to reduce dimensionality, HDBSCAN for clustering, and c-TF-IDF to form a representation of topics [3]. BERTopic demonstrates higher topic coherence compared to LDA, especially on short or variable texts. A study [14] noted that embedding models (e.g., BERTopic or Combined Topic Model CTM) can outperform LDAs in terms of NPMI/coherence, but require more computational resources. In addition, studies on Indo-Aryan languages (e.g. Hindi) show that BERTopic consistently outperforms classical methods on short texts [15].

Evaluation of thematic models is carried out through metrics such as perplexity, coherence (UMass, Cv) and NPMI. The reviews emphasise that reducing perplexity does not guarantee an increase in coherence, so a combined approach is recommended [11-12]. There are additional challenges for the corpora of the Ukrainian language: rich inflexion, word formation, morphological variants, and a lack of large, composed datasets. For example, in the study of Podillya folklore, the need for lemmatisation and morphological normalisation before the use of LDA is emphasised [6]. Thus, it is essential for Ukrainian-speaking buildings to take into account:

- Careful pre-processing of the text (lemmatisation, correction of inflexions),
- Choice of model depending on the genre (long formats – LDA/NMF, short/social texts – BERTopic/CTM),
- Multi-metric evaluation (perplexy and NPMI/coherence) and, if possible, human validation.

Thus, the literature demonstrates the evolution of thematic modelling [16-21]: from traditional generative models to modern approaches with contextual embeddings. For the analysis of Ukrainian-language texts, it is advisable to use a hybrid strategy: to use LDA as a basic model for long documents, and for short/noisy texts – models on embeddings; At the same time, to ensure high-quality pre-processing and combined assessment. In the following sections, these recommendations will be taken into account when choosing a model, adjusting hyperparameters, and evaluating the results.

Within the framework of this study, the construction of the thematic model was carried out on the basis of a probabilistic approach implemented through Latent Dirichlet Allocation (LDA). The LDA algorithm assumes that each document in the corpus is a mixture of several topics, and each topic is a distribution of word probabilities. Thus, the mathematical essence of the model is to restore the hidden parameters of these distributions. Let:

- $D=\{d_1, d_2, \ldots, d_M\}$ – a corpus of documents consisting of M documents;
- $V=\{w_1, w_2, \ldots, w_N\}$ is a dictionary that contains N of unique words;
- K is the number of hidden topics.

Each document d_m is modelled as a stochastic process of generating words according to the following steps:

1. For each document d_m, the distribution of topics is chosen $\theta_m \sim \mathrm{Dir}(\alpha)$, where α is a hyperparameter of the Dirichlet distribution that controls the "blurring" of topics in the document.

2. For each k topic, the distribution of words is determined $\phi_k \sim \mathrm{Dir}(\beta)$, where β is a hyperparameter that controls the "blurredness" of words in the subject.

3. For each word w_{mn} – in document d_m:

- The topic $z_{mn} \sim \mathrm{Mult}(\theta_m)$;
- Next, a word on this topic is chosen: $w_{mn} \sim \mathrm{Mult}(\phi_{z_{mn}})$.

The total probability of the body of documents is given as:

$$P(W, Z|\alpha, \beta)=\prod_{m=1}^{M} \int P(\theta_m|\alpha)\left(\prod_{n=1}^{N_m} \sum_{z_{mn}} P(z_{mn}|\theta_m) P(w_{mn}|z_{mn}, \beta)\right) d\theta_m, \quad (1)$$

where W are all observed words, Z are hidden variables (topics for each word).

The purpose of thematic modelling is to find a posteriori distribution:

$$P(\theta, \phi, Z|W, \alpha, \beta), \quad (2)$$

which is approximated using the variational Bayesian approach or the Gibbs sampling method.

For each document, the following are calculated:

- $\theta_m=(\theta_{m1}, \theta_{m2}, \ldots, \theta_{mK})$ – is the probability vector of topics;
- $\phi_k=(\phi_{k1}, \phi_{k2}, \ldots, \phi_{kN})$ – is the probability vector of words in the subject.

After training the model, each paper is presented as a combination of topics with weights – θ_m, and each topic is defined by the most likely words from the vector ϕ_k.

Two primary metrics evaluate the quality of thematic modelling:

1. Perplexy is a measure of the consistency of the model with the test data:

$$\text{Perplexity}\left(D_{test}\right)=\exp\left\{\frac{-\sum_{d\in D_{test}}\log P\left(w_d\right)}{\sum_{d\in D_{test}}N_d}\right\}. \quad (3)$$

A lower perplexy value corresponds to better model consistency.

2. Topic Coherence – assesses the semantic consistency of the most important words of the topic. For a set of words $W_t=\{w_1,w_2,\ldots,w_M\}$, coherence is defined as:

$$C\left(W_t\right)=\sum_{i<j}\log\frac{D\left(w_i,w_j\right)+\epsilon}{D\left(w_j\right)}, \quad (4)$$

where $D\left(w_i,w_j\right)$ – is the number of documents in which the words w_i and w_j – occur together, ϵ is the smoothing factor.

The result of modelling is a set of thematic distributions:

$$\Phi=\{\phi_1,\phi_2,\ldots,\phi_K\},\Theta=\{\theta_1,\theta_2,\ldots,\theta_M\}, \quad (5)$$

which allows the construction of the matrix $M\times K$, where each element of θ_{mk} – is interpreted as the probability that the document d_m – belongs to the topic k.

This matrix forms the basis for further analysis – clustering of documents, construction of semantic maps, and visualisation of thematic structures.

The goal of the developed software product is to create an efficient, automated system for thematic modelling of texts, which allows the user to quickly identify key topics in unstructured text documents without prior mark-up or classification. This tool should provide the ability to:

1. Process large amounts of textual information;
2. Analyse the content of documents without manual intervention;
3. Identify hidden topics by applying machine learning methods, in particular the Latent Dirichlet Allocation (LDA) algorithm;
4. Display the results in an understandable, interpreted form – in the form of lists of keywords that form topics, and their distribution in documents;
5. Visualise the results of thematic analysis to improve understanding of the structure of the text corpus.

As a result of the development, a software module will be implemented that helps the user interpret large arrays of texts, reduce the time for their processing, and identify semantic trends in the content without deep linguistic or technical preparation.

A software product for thematic modelling of texts should implement a complete cycle of automated analysis of textual information, from downloading data to displaying results in a convenient form. The main functions that the system should implement include:

1. Loading the Text Corpus:

- Load input text data from a file (e.g., .txt, .csv,. json).
- Support for entering one or more documents.
- Ability to work with Ukrainian-language texts.

2. Pre-processing of texts:

- Clearing punctuation, numbers, and special characters.
- Lowercase texting.
- Delete stop words (in Ukrainian).

- Tokenisation is the division of text into separate words (tokens).
- Lemmatisation (if necessary) – bringing words to their original form.

3. Building a thematic model:

- Formation of a vector representation of texts (for example, Bag-of-Words or TF-IDF).
- Building an LDA model to define topics in the corpus.
- Setting the number of topics, dictionary sizes, alpha and beta parameters.

4. Generate Results:

- Create lists of topics with a set of keywords for each.
- Calculate the distribution of topics for each document.
- Save results in text or tabular format.

5. Visualisation of results:

- Building an interactive thematic map (through the pyLDAvis library).
- Displaying the weights of words in topics.
- Visualise similarities and overlaps between topics.

6. Data Saving/Export:

- Option to export model, topic lists, or topic breakdowns to a file.
- Ability to reuse the saved model.

7. User Interface (Optional). A simple graphical interface (or console menu) where the user can:

- select a file;
- set model parameters (number of topics);
- View visualisation.

Thus, the software product should cover all the key stages of thematic analysis of texts – from processing and modelling to interpretation and output of results.

The developed software product is aimed at users who need to analyse a large amount of textual information, but do not have sufficient technical or linguistic knowledge for its deep processing. Thanks to the automation of the main processes of thematic analysis, the system allows you to solve a number of applied problems.

1. Analysis of a large volume of texts. In the real world, processing thousands of documents manually is extremely time-consuming and resource-intensive. The software product allows you to automatically process large corpora of texts without the need for human intervention at each stage.

2. Automatic detection of topics in documents. The user gets the opportunity to find out what the texts are about, even if the issues have not been determined in advance. The LDA-based model allows you to automatically generate topics based on statistical patterns in the data.

3. Classification and grouping of texts by topic. The system allows you to determine which documents are related to a particular topic, which makes it possible to segment texts by content (for example: economics, politics, sports, culture).

4. Building thematic profiles of documents. Each document has a distribution of topics that helps to assess which topics dominate the text and which are secondary. It is beneficial for reports, news, research articles, or social media content.

5. Visualisation of results for analytics. The user receives an interactive visualisation (via pyLDAvis), which allows a better understanding of the structure of topics, their relationships, and their distribution in the space of texts. It makes the analysis accessible even to non-professional users.

6. Decision support. Thanks to the interpretation of the results of thematic modelling, the user can faster identify trends, filter important documents and to draw conclusions based on the actual content of the texts.

7. Saving the results for further processing. The resulting topics and breakdowns can be stored, used in other systems, or used to generate reports, making the product useful in research, educational, and business contexts.

Thus, the software product removes the need for the user to manually process texts and allows you to obtain valuable semantic insights automatically, quickly and in a convenient format.

The system being developed, conventionally called the "Thematic Analysis Module", is a software tool for processing, analysing and visualising large volumes of text documents in order to automatically identify semantic topics. The system belongs to the application software that implements computational linguistics and machine learning methods for the needs of text analysis. The principle of operation of the module is based on the Latent Dirichlet Allocation (LDA) algorithm, one of the most common approaches to thematic modelling, which allows you to determine a set of hidden topics in the corpus of texts without human intervention.

1. What actions take place on the input data? After loading the text corpus, the system performs several sequential stages of processing on the input data to prepare it for thematic analysis:

– Text pre-processing:

- Clearing texts from punctuation, memorable characters, and numbers.
- Normalisation (converting all words to lower case).
- Remove stop words that do not carry a semantic load (for example, and, or, also).
- Tokenisation is the division of text into separate words (tokens).
- Lemmatisation is the reduction of words to their original form (for example: "worked" → "work").

– Building a textual representation:

- Creating a document-term matrix using Bag-of-Words or TF-IDF methods;
- Formation of a dictionary (all unique words of the corpus).

– Thematic modelling:

- Building an LDA model – automatic detection of topics that are repeated in texts;
- Definition of the set of words that best characterise each topic;
- Calculate the distribution of topics in each document (i.e. which document is about what).

2. What the user sees at the output. After completing all stages of processing and modelling, the system provides the user with the result in a clear and visualised form:

–Theme. A list of topics was found, each of which is represented by a set of keywords with the highest weight (importance). For example:

Topic 1: ['economy', 'profit', 'currency', 'inflation', 'bank']

Topic 2: ['sport', 'match', 'team', 'goal', 'tournament']

– Distribution of topics by documents. Each document displays which topics dominate it and with what probability. For example: Document No. 7: Topic 1 – 60%, Topic 3 – 30%, Topic 5 – 10%.

– Interactive visualisation. Using the pyLDAvis library, the results are displayed as an interactive topic map where:

- circles are themes;
- circle size – frequency of the topic;
- overlap – similarity between topics;
- You can hover the cursor and see the topic keywords.

– Tables with results. Export in CSV or JSON formats:

- table with topics;
- distribution of topics by documents;
- Top words for each topic.

The user can use all these results to:

- building reports;
- content analytics;
- segmentation of texts by topic;
- automatic classification or preparation of training data.

First of all, it must have the functionality to load the corpus of documents in a convenient format (for example, TXT or CSV), which can contain both individual texts and large arrays of text data, tokenisation, deletion of stop words, and, if necessary, also lemmatisation to bring words to the basic form. Based on the cleaned texts, the program should build a thematic model using the Latent Dirichlet Allocation (LDA) method, which allows you to automatically determine a set of topics in a given corpus, each of which will be represented by a set of keywords. Once processed, the results must be stored in a file or database, allowing them to be reused, exported, or integrated with other systems. Finally, the system should display the results to the user clearly and intuitively – both in the form of lists of topics and tables, and in the form of interactive visualisation of issues, which significantly simplifies the analysis of the data obtained.

The project aims to create an effective tool for automatic thematic analysis of texts, which allows you to identify content structures in unstructured text data without the need for their preliminary labelling. The goal of the project is to provide the user with an accessible tool for analysing large corpora of texts, with the ability to visualise, store and interpret the results without deep technical knowledge. The Python programming language was chosen as the implementation environment of the software product using the libraries Gensim (for building a thematic model of LDA) and pyLDAvis (for visualising the results). Additionally, the NLTK, Pandas, and Matplotlib libraries can be used to process texts and present results. The program provides an interface in the form of a CLI (command line) or, by extension, a simple graphical interface (GUI) based on the Tkinter or Streamlit libraries, which allows you to interact with the user conveniently - select a file, set analysis parameters, and start processing. Among the main limitations of the product are the focus on texts in Ukrainian (for correct lemmatisation and a list of stop words), as well as the need for a pre-prepared document format (for example, one document – one line in a file). There are two types of users: the regular user, who runs the analysis, views the results, and exports them in a convenient format, and the administrator or developer, who can change model settings, update dictionaries, or change the architecture of the module for a specific application. Scripts:

1. Download the document corpus – the user imports a set of texts for analysis.
2. Configure model – sets the parameters of thematic modelling (number of topics, type of filtering).
3. Run simulation – the system pre-processes and builds an LDA model.
4. View results – the user receives a list of topics, keywords, and breakdowns by documents.
5. Save results – the results are exported to a file (CSV, JSON, etc.).

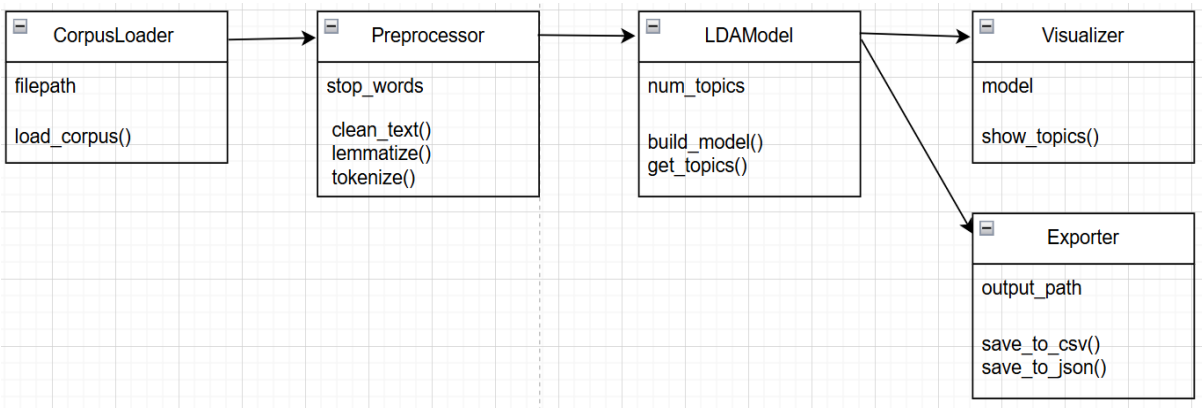

Figure 1: Class Diagram

A class diagram displays the structure of a system, that is, what classes it consists of, what functions each class performs, and how these classes are related to each other. It answers the question: "What parts are included in the program and what do they do?". The main classes in the diagram are:

- CorpusLoader – Responsible for loading text data.
- Pre-processor – cleans, tokenises, and lemmatises texts.
- LDAModel builds a thematic model, manages learning, and produces results.
- Visualizer – creates a visualisation of topics (e.g. via pyLDAvis).
- Exporter – saves the results to a file.
- Links:
- Each class has one or more methods, which are displayed at the bottom of the rectangle.
- The arrows between classes show dependencies: for example, LDAModel uses Pre-processor to prepare data, and Visualizer uses it to plot based on model results.

This diagram is needed to describe the architecture of the code and helps the programmer to implement the system correctly. The sequence diagram shows the order in which actions are performed in time – that is, what exactly happens in the system from the start of the start-up to the receipt of the result. It simulates how objects pass requests to each other, including in what order. Sequence of events:

1. The user gives a command through the interface to start the analysis.
2. CorpusLoader loads data.
3. Pre-processor cleans and processes texts.
4. LDAModel performs thematic modelling.
5. Visualizer displays themes on the screen.

Exporter allows you to save the results. Each object has a vertical "lifeline", and arrows show the interactions between them. The lower on the diagram, the later in time the action takes place. This diagram gives an idea of the logic of executing a program step by step and is very useful for testing or scripting.

During the work, it was determined that the proposed system – the "Thematic Analysis Module" – is designed to automatically process large volumes of unstructured text in order to identify key topics without preliminary data labelling. This approach is especially relevant in today's information environment, where a vast number of text messages are generated every day, which require quick and meaningful analysis. In the process of formalisation, the input and output data of the system, the main stages of word processing (cleaning, tokenisation, lemmatisation), building a

model and displaying the results were described. It is established that the system should provide the loading of the body of documents, the adjustment of model parameters, the execution of simulations, the visualisation of the results and the ability to export the received data. All these features have been described in the form of functional requirements for the product. In order to structurally present the work of the module, a technical task was created, which describes in detail the implementation environment (Python, Gensim, pyLDAvis), user interaction interface, target audience (ordinary user, analyst) and system limitations (focus on Ukrainian-language texts). Particular attention is paid to the visualisation of the system architecture using UML diagrams. Created:

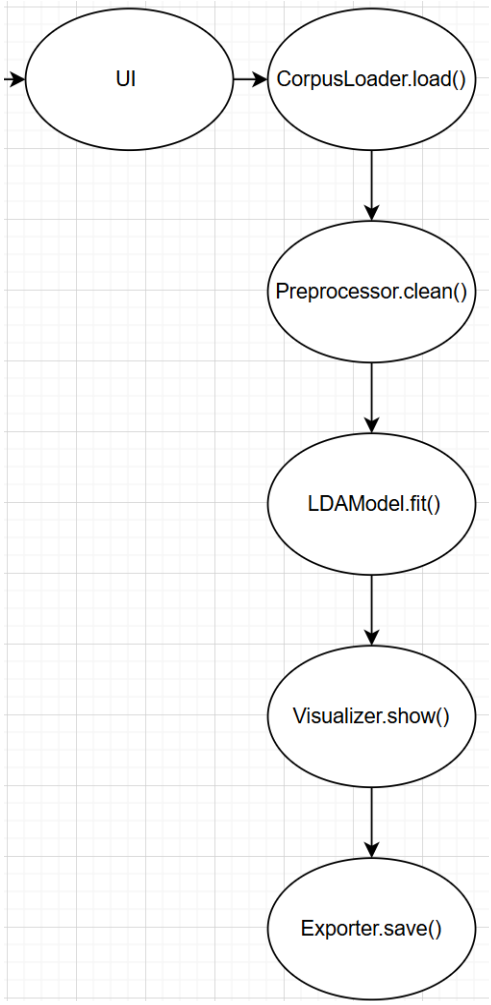

Figure 2: Sequence Diagram

- a class diagram showing the internal structure of the program and the relationships between its modules (CorpusLoader, Pre-processor, LDAModel, Visualizer, Exporter);
- Sequence Diagram, which illustrates the step-by-step process of performing thematic analysis – from loading texts to saving results.

The work done made it possible to systematise the idea of the logic of the software product's functioning, to determine the key elements and their interaction. The results obtained from a solid basis for further development, testing, and implementation of thematic modelling in real text analysis tasks. Thus, the work has been completed, and the goals set – to formulate the requirements, build the architecture and model the system – have been achieved in full.

The task of thematic modelling is to find latent (hidden) topics in a large corpus of text documents without predefined labels. Formally, each document d_m from the corpus

$D=\{d_1,d_2,\ldots,d_M\}$ is considered as a stochastic mixture of several topics $T=\{t_1,t_2,\ldots,t_K\}$, where K is the number of topics determined by the user or selected empirically. Let the dictionary of all unique terms that occur in the corpus be denoted as $V=\{w_1,w_2,\ldots,w_N\}$, where N is the number of unique words. Each d_m document can then be submitted as a sequence of tokens:

$$d_m=(w_{m1},w_{m2},\ldots,w_{mN_m}), \tag{6}$$

where N_m – is the number of words in the document d_m.

The central hypothesis of the model is that documents are created as a result of a stochastic process, in which topics are first chosen, and then words inherent in these topics. Thus, the document does not belong to one topic, but has a specific distribution of probabilities of topics, which reflects the content structure of the text. To model this process, two groups of latent variables are introduced:

- $\theta_m=(\theta_{m1},\theta_{m2},\ldots,\theta_{mK})$ – is the distribution of topics in the document d_m;

$$\theta_m\sim\text{Dirichlet}(\alpha), \tag{7}$$

where $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_K)$ – hyperparameters that determine the concentration of the distribution (the smaller the value of α_k, the more the document gravitates towards one topic).

- $\phi_k=(\phi_{k1},\phi_{k2},\ldots,\phi_{kN})$ – is the distribution of words for each topic t_k;

$$\phi_k\sim\text{Dirichlet}(\beta), \tag{8}$$

where $\beta=(\beta_1,\beta_2,\ldots,\beta_N)$ – are hyperparameters that determine the "blurredness" of the topic in relation to a set of words.

For each word w_{mn} in the document d_m, the generation process is described as follows:

1. Select the topic $z_{mn}\sim\text{Multinomial}(\theta_m)$;
2. Choose the word $w_{mn}\sim\text{Multinomial}(\phi_{z_{mn}})$.

Thus, the probability of generating a document d_m is defined as:

$$P(d_m|\alpha,\beta)=\int P(\theta_m|\alpha)\left(\prod_{n=1}^{N_m}\sum_{z_{mn}=1}^{K}P(z_{mn}|\theta_m)P(w_{mn}|z_{mn},\beta)\right)d\theta_m, \tag{9}$$

and for the entire body:

$$P(D|\alpha,\beta)=\prod_{m=1}^{M}P(d_m|\alpha,\beta). \tag{10}$$

The inputs of the model are D document corpus, glossary of V terms, hyperparameters α,β and number of K themes. The initial data are:

- Matrix of distribution of topics by documents $\Theta=[\theta_m]_{M\times K}$;
- Matrix of word distribution by topics – $\Phi=[\phi_k]_{K\times N}$;
- Set of latent variables $Z=\{z_{mn}\}$, which determines which word belongs to which topic.

The purpose of thematic modelling is to find the maximum a posteriori estimation of the parameters Θ and Φ based on the observed data of the W:

$$\widehat{\Theta},\widehat{\Phi}=arg\max_{\Theta,\Phi} P(\Theta,\Phi|W,\alpha,\beta), \tag{11}$$

which is equivalent to maximising the probability of the case:

$$P(W|\alpha,\beta)=\prod_{m=1}^{M}\int P(\theta_m|\alpha)\left(\prod_{n=1}^{N_m}\sum_{z_{mn}} P(z_{mn}|\theta_m)P(w_{mn}|z_{mn},\beta)\right)d\theta_m. \tag{12}$$

Since the direct estimation of this expression is computationally complex, approximation methods are used in practical implementation – in particular:

- Variational Bayesian approach, which reduces the problem to optimising the lower limit of the logarithm of plausibility;
- Gibbs Sampling is a stochastic method of estimating parameters using samples from conditional distributions.

After training the model, each document d_m – is presented as a vector of topics:

$$d_m=[\theta_{m1},\theta_{m2},\ldots,\theta_{mK}], \tag{13}$$

And each topic t_k – is described by a set of most likely words:

$$t_k=\{w_i\in V:\phi_{ki}\,\text{maximum}\}. \tag{14}$$

Thus, the task of thematic modelling is reduced to finding such parameters Θ and Φ, which best explain the observed texts, i.e. maximise the total probability of $P(W,Z|\alpha,\beta)$. This formalisation creates the basis for further implementation of the system of thematic analysis, assessment of the quality of the model and interpretation of the content of the obtained topics.

Thematic modelling using the Latent Dirichlet Allocation (LDA) algorithm is based on a probabilistic generative model that describes how texts may have been created from a particular set of topics. The key idea is that the process of writing each paper can be thought of as a random process of selecting topics and words from those topics, driven by statistical patterns of co-occurrence of tokens. LDA models each document as a stochastic mixture of topics, and each topic as a stochastic mixture of words. It means that:

- The document does not belong to only one topic, but contains particles of several issues.
- Different topics with specific probabilities can generate a word in a document.

Thus, for each document d_m (where $m=1,2,\ldots,M$) and each word w_{mn} in it ($n=1,2,\ldots,N_m$), the model describes a two-stage generation process:

1. Choosing a topic z_{mn} for the word w_{mn}, according to a multivariate distribution:

$$z_{mn}\sim\text{Multinomial}(\theta_m), \tag{15}$$

where θ_m – is the probability vector of topics for the document d_m.

2. Choosing the word w_{mn} from the distribution of words of the topic $\phi_{z_{mn}}$:

$$w_{mn}\sim\text{Multinomial}(\phi_{z_{mn}}), \tag{16}$$

where $\phi_{z_{mn}}$ – is the probability vector of words for the topic z_{mn}.

The LDA model uses the Dirichlet distribution, a multivariate generalisation of the beta distribution, which is often used as a prior distribution for probability quantities having the sum of 1. For each document, the parameter θ_m – is selected as:

$$\theta_m \sim \text{Dirichlet}(\alpha), \tag{17}$$

where α is a hyperparameter vector that controls the level of concentration of topics in the document:

- If $\alpha_k < 1$, the document contains a small number of dominant topics;
- If $\alpha_k > 1$, the document has a more uniform combination of issues.

Similarly, for each topic t_k, the parameter ϕ_k – is selected as:

$\phi_k \sim \text{Dirichlet}(\beta)$,

where β determines the degree of "blurredness" of the distribution of words in the subject:

- small $\beta \longrightarrow$ each topic is concentrated around a few keywords;
- large $\beta \longrightarrow$ words are distributed more evenly.

Given the dependencies, it is possible to write the common probability for all variables of the model:

$$P(W, Z, \Theta, \Phi | \alpha, \beta) = \prod_{k=1}^{K} P(\phi_k | \beta) \prod_{m=1}^{M} P(\theta_m | \alpha) \prod_{n=1}^{N_m} P(z_{mn} | \theta_m) P(w_{mn} | z_{mn}, \Phi), \tag{18}$$

where $W = \{w_{mn}\}$ – all corpus words; $Z = \{z_{mn}\}$ – hidden topics for each word; $\Theta = \{\theta_m\}$ – topic distributions for all documents; $\Phi = \{\phi_k\}$ – word distributions for all topics.

To obtain the probability of the observed data (i.e. words), you need to integrate for all latent variables:

$$P(W | \alpha, \beta) = \int_{\Theta} \int_{\Phi} \sum_{Z} P(W, Z, \Theta, \Phi | \alpha, \beta) d\Theta d. \tag{19}$$

This expression is the fundamental equation of thematic modelling, which describes how well specific parameters Θ and Φ explain the corpus of texts. The goal of thematic analysis is to find a posteriori distribution of hidden variables:

$$P(\Theta, \Phi, Z | W, \alpha, \beta) = \frac{P(W, Z, \Theta, \Phi | \alpha, \beta)}{P(W | \alpha, \beta)}. \tag{20}$$

However, an accurate calculation of this distribution is not possible due to the large number of parameters and complex integration. Therefore, approximation estimation methods are used:

- Variational Bayesian method (VB) replaces the actual distribution with an approximate one, minimising the Kullback–Leibler divergence between the two.
- Collapsed Gibbs Sampling: Repeatedly generates Z samples from conditional distributions, gradually approximating the actual distribution.

In conditional form, Gibbs sampling allows you to update the topic for each word – w_{mn}, using the formula:

$$P\left(z_{mn}=k\middle|Z_{\neg mn},W,\alpha,\beta\right)\propto\left(n_{m,k}^{\neg mn}+\alpha_k\right)\frac{n_{k,w_{mn}}^{\neg mn}+\beta_w}{n_k^{\neg mn}+\sum_{w'}\beta_{w'}}, \tag{21}$$

where $n_{m,k}^{\neg mn}$ – the number of words in the mmm document related to the topic k (without the current word); $n_{k,w_{mn}}^{\neg mn}$ – the number of times the word w_{mn} appeared in the topic k; $n_k^{\neg mn}$ – is the total number of words attributed to the topic k; α_k,β_w – are Dirichlet hyperparameters.

This expression reflects the interdependence between topics and words: a word with a higher frequency in a specific topic is more likely to be reassigned to it. After the algorithm converges, two sets of parameters are obtained:

- matrix Θ (size $M\times K$) – displays the distribution of topics for each document;
- matrix Φ (size $K\times N$) – describes the probabilities of words in each topic.

These matrices form the basis for:

- building thematic profiles of documents;
- visualisations in the form of thematic maps (*pyLDAvis*);
- quantitative analysis using coherence and perplexy metrics.

The LDA probabilistic model combines mathematical accuracy and practical interpretation. Unlike the "black boxes" of neural networks, its parameters have a clear semantic interpretation:

- ϕ_k – is the verbal core of the topic, whose keywords can describe.
- θ_m – is a thematic imprint of the document, showing which topics dominate it.

Thus, LDA not only formally classifies texts, but also reproduces the hidden content structure of the language, which makes it an indispensable tool in the tasks of analytics, knowledge search and classification of large text corpora.

The parameter estimation process in the *Latent Dirichlet Allocation (LDA)* model is to restore the a posteriori parameter distributions $\Theta=\{\theta_m\},\Phi=\{\phi_k\}$ and hidden variables $Z=\{z_{mn}\}$, which are as consistent as possible with the available data of the corpus $W=\{w_{mn}\}$. Since the direct analytical calculation of the posterior distribution $P(\Theta,\Phi,Z|W,\alpha,\beta)$ – is impossible due to the complexity of integrals and sums, approximation methods of estimation are used, which give statistically justified approximations. Recall that the posterior distribution in the LDA model is as follows:

$$P(\Theta,\Phi,Z|W,\alpha,\beta)=\frac{P(W,Z,\Theta,\Phi|\alpha,\beta)}{P(W|\alpha,\beta)}. \tag{22}$$

Here, the denominator $P(W|\alpha,\beta)$ – is the normalising integral:

$$P(W|\alpha,\beta)=\int_\Theta\int_\Phi\sum_Z P(W,Z,\Theta,\Phi|\alpha,\beta)\,d\Theta\,d, \tag{23}$$

Therefore, stochastic and variational methods are used to calculate the parameters of the model, which allow you to approximate this distribution.

The Variational Bayes (VB) method is based on the idea of replacing a complex a posteriori distribution with a simpler one that belongs to the parameterised family of Q distributions. Let $Q(\Theta,\Phi,Z)$ – is an approximation distribution that tries to get as close as possible to the true $P(\Theta,\Phi,Z|W,\alpha,\beta)$. The measure of proximity between them is measured using the Dandelion–Leibler divergence (KL-divergence):

$$KL(Q \vee ¿ P) = \int Q(\Theta, \Phi, Z) \log \frac{Q(\Theta, \Phi, Z)}{P(\Theta, \Phi, Z | W, \alpha, \beta)} d\Theta \, d\Phi. \quad (24)$$

The purpose of the variational method is to minimise the KL divergence, i.e. to find the Q parameters that best approximate the actual distribution. It is equivalent to maximising the lower limit of the logarithm of plausibility (ELBO – Evidence Lower Bound):

$$L(Q) = E_Q[\log P(W, Z, \Theta, \Phi | \alpha, \beta)] - E_Q[\log Q(\Theta, \Phi, Z)], \quad (25)$$

which is the lower estimate of the log plausibility of the data $\log P(W | \alpha, \beta) \geq L(Q)$.

The optimisation process $L(Q)$ – is performed iteratively to convergence, similar to the Expectation-Maximisation (EM) algorithm. An alternative and more intuitive method is Gibbs sampling, which belongs to the class of Markov Chain Monte Carlo (MCMC) methods.

The main idea is to sequentially choose topics for each word based on conditional probability:

$$P(z_{mn} = k | Z_{\neg mn}, W, \alpha, \beta) \propto (n_{m,k}^{\neg mn} + \alpha_k) \frac{n_{k,w_{mn}}^{\neg mn} + \beta_{w_{mn}}}{n_k^{\neg mn} + \sum_{w'} \beta_{w'}}, \quad (26)$$

where $n_{m,k}^{\neg mn}$ – is the number of words in the mmm document belonging to the subject of k (except for the current word); $n_{k,w_{mn}}^{\neg mn}$ – is the number of times the word w_{mn} appears in the topic k; $n_k^{\neg mn}$ – is the total number of words in the subject of k; $\alpha_k, \beta_{w'}$ – are hyperparameters of Dirichlet distributions.

After many iterations of sampling, the circuit reaches a steady state in which the Z samples are distributed according to an accurate posteriori distribution.

$$\widehat{\theta_{m,k}} = \frac{n_{m,k} + \alpha_k}{\sum_{k'} (n_{m,k'} + \alpha_{k'})}, \quad \widehat{\phi_{k,w}} = \frac{n_{k,w} + \beta_w}{\sum_{w'} (n_{k,w'} + \beta_{w'})}. \quad (27)$$

The α, β hyperparameters determine the structure and coherence of topics and can be fixed (pre-selected empirically) or optimised when training the model.

Optimisation can be performed using the maximum likelihood (ML) method or posteriori estimate maximisation (MAP α).

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} - \frac{g(\alpha_k)}{h(\alpha_k)}, \quad (28)$$

where $g(\alpha_k)$ and $h(\alpha_k)$ – is the gradient and the second derivative (Hessian) of the log-probability function.

Small values of α lead to the dominance of a few topics in the document, while large values lead to a more even distribution. Similarly, a small β – causes topics to concentrate around a limited number of words, while a significant β – increases the confusion of issues.

LDA training continues until the values of the target function or metric (e.g., ELBO or log plausibility) stabilise:

$$\left| L^{(t+1)} - L^{(t)} \right| < \varepsilon, \quad (29)$$

where ε is the stop threshold (usually 10^{-4} or 10^{-5}). In the case of Gibbs sampling, convergence is checked by the stability of the estimates of the parameters θ and ϕ over several successive iterations.

Upon completion of the evaluation process, the following results are obtained:

- The matrix $\Theta=\left[\theta_{m,k}\right]$ – describes which topics are present in each document and with what probability.
- The matrix $\Phi=\left[\phi_{k,w}\right]$ – shows how characteristic a specific word is for a particular topic.
- Vector Z – displays the thematic affiliation of each word of the corpus. It information can be used to build thematic maps and frequency analytics.

The obtained parameters have a clear statistical and substantive interpretation $\theta_{m,k}$ – is the probability that the document d_m belongs to the topic t_k; $\phi_{k,w}$ – is the probability that the word www is representative of the topic t_k. Based on these parameters, you can:

- To determine the dominant themes of texts.
- Analyse the thematic structure of the corpus.
- Build semantic connections between topics.
- Apply the results for further machine learning (classification, search, recommendation systems).

Since thematic modelling is an unsupervised method, classical classification metrics (accuracy, F1, etc.) are not suitable. Therefore, the quality of the Latent Dirichlet Allocation (LDA) model is assessed by internal metrics – those based on logical, probabilistic or semantic criteria. The main ones are Perplexity, Coherence and its improved version NPMI (Normalised Pointwise Mutual Information). Perplexity is a classic probabilistic measure that assesses how well a topic model agrees with test data. Intuitively, it measures the "degree of surprise" of the model when trying to reproduce new texts: the smaller the perplexity value, the better the model reflects the structure of the language. Let's say we have a test corpus of documents – D_{test}, which was not used when training the model. Then perplexity is defined as:

$$\text{Perplexity}\left(D_{test}\right)=\exp\left\{-\frac{\sum_{d=1}^{M_{test}} log\, P\left(w_d\right)}{\sum_{d=1}^{M_{test}} N_d}\right\}, \tag{30}$$

where $P\left(w_d\right)$ – is the probability of a d document according to the LDA model; N_d – the number of words in the document; M_{test} – is the number of documents in the test case.

The probability of a d document is calculated as:

$$P\left(w_d\right)=\prod_{n=1}^{N_d}\sum_{k=1}^{K} P\left(w_{dn}\middle|z_{dn}=k\right)P\left(z_{dn}=k\middle|d\right). \tag{31}$$

- A smaller perplexity value ⟶ the model is better at predicting unknown texts.
- If perplexity decreases with the number of topics, but then begins to grow, this indicates retraining.
- Typical values for well-balanced Ukrainian-language corpora (with purification and lemmatisation):
  a. $\text{Perplexity}\approx 600-900$ – low quality, blurry themes;
  b. $\text{Perplexity}\approx 300-500$ – satisfactory quality;
  c. $\text{Perplexity}\approx 100-250$ – high model consistency.

For a corpus of 5000 Ukrainian news articles, after lemmatisation and elimination of stop words, the following was obtained:

Table 4
Results of the analysis of the corpus of 5000 Ukrainian news articles

Number of Topics (K)	Perplexy ↓
5	720.3
10	480.6
15	290.2
25	275.8
40	310.1

The optimal is $K=15-25$, where there is a minimum of perplexy; therefore, the model is best consistent with the linguistic structure of the texts.

Coherence is a semantic metric that assesses how meaningful the topics formed by the model are. It measures how often the most important words of the topic occur together in texts, that is, whether there is a logical connection between them. For each topic t_k defined by the set of the most important words $W_k=\{w_1, w_2, \ldots, w_M\}$, the coherence is calculated as:

$$C_{UMass}(W_k)=\frac{2}{M(M-1)}\sum_{i<j}\log\frac{D(w_i,w_j)+\epsilon}{D(w_j)}, \tag{32}$$

where $D(w_i,w_j)$ – is the number of documents in which the words w_i and w_j occur together; $D(w_j)$ – is the number of documents containing the word w_j; ϵ – is a small number to prevent division by zero.

The coherence of the entire model is calculated as an average of all topics:

$$C_{model}=\frac{1}{K}\sum_{k=1}^{K}C_{UMass}(W_k), \tag{33}$$

where $C>0$ – is the topic is semantically consistent; $C\approx 0$ – is a weak connection between words; $C<0$ – is a random or chaotic topic.

For Ukrainian texts, in particular, news or popular science corpora, the following values are acceptable:

- $C_{model}\in[0.3;0.6]$ – is good coherence;
- $C_{model}>0.6$ – is high semantic consistency.

For a model with 15 topics built on a news corpus, the following coherences are obtained:

Table 5
Results of the analysis of the 15-topic model built on the news corpus

No.	Top Words	C_{UMass}
1	{government, budget, economy, taxes, finance}	0.61

2	{war, army, front, soldiers, weapons}	0.68
3	{education, students, university, science, training}	0.54
4	{sport, match, team, championship, players}	0.62
…	…	…

The average coherence of the model is $C_{model}=0.61$, which indicates good thematic separation and high content of topics. The NPMI (normalised mutual information) metric is a modern improvement of coherence that takes into account the statistical interdependence of words in a topic. For a pair of words (w_i, w_j) – is defined:

$$\mathrm{PMI}(w_i, w_j)=\log\frac{P(w_i, w_j)}{P(w_i)P(w_j)}, \tag{34}$$

and the normalised form:

$$\mathrm{NPMI}(w_i, w_j)=\frac{\mathrm{PMI}(w_i, w_j)}{-\log P(w_i, w_j)}. \tag{35}$$

Then the coherence of the topic is calculated as:

$$C_{NPMI}(W_k)=\frac{2}{M(M-1)}\sum_{i<j}\mathrm{NPMI}(w_i, w_j), \tag{36}$$

and for the entire model:

$$C_{NPMI}^{model}=\frac{1}{K}\sum_{k=1}^{K}C_{NPMI}(W_k), \tag{37}$$

where $C_{NPMI}\in[0,1]$. The closer to 1, the better the semantic matching; a value of ¿0.2 indicates noise or uninterpreted topics. For the Ukrainian News Corpus (≈10,000 documents, 200,000 lem):

Table 6
Results of the analysis of the Ukrainian News Corpus (≈10,000 documents, 200,000 lems)

Number of Topics (K)	$C_{NPMI}^{model}\uparrow$
5	0.21
10	0.37
15	0.48
20	0.46
30	0.42

The optimal model with $K=15$ topics has $C_{NPMI}^{model}=0.48$, which corresponds to high coherence for Ukrainian-language texts.

Table 7
Benchmarking metrics

Metric	Grade Type	Optimum	Interpretation
Perplexy	Probabilistic	↓ Minimum	Less value ⟶ better model consistency
Coherence (UMass)	Semantic	↑ maximum	Topics have logically related words
NPMI	Normalized semantic	↑ maximum	Quantifies meaningfulness and stability of topics

For medium-sized Ukrainian-language buildings (3-10 thousand documents), the best balance is observed when:

- $K \in [10,20]$,
- $\text{Perplexity} \approx 250-400$,
- $C_{UMass} \in [0.4,0.6]$,
- $C_{NPMI}^{model} \in [0.45,0.55]$.

The perplexity, coherence, and NPMI metrics complement each other:

- Perplexy evaluates the statistical consistency of the model.
- Coherence – semantic logic.
- NPMI – normalised content of topics in the context of the entire language.

For Ukrainian-language corpora, where linguistic morphology is more complex (cases, prefixation, suffixation), the best results are achieved after lemmatisation, elimination of stop words, and normalisation of frequencies before building LDA. The combined use of metrics allows not only to assess the quality of the model, but also to select the optimal number of K topics and hyperparameters α, β, providing a balance between accuracy and interpretation of the results.

The parameters obtained as a result of training *the Latent Dirichlet Allocation (LDA)* model have both statistical and semantic significance. After performing the evaluation procedures, the model is defined by two main matrices:

- The matrix $\Theta=[\theta_{m,k}]_{M\times K}$ – reflects the probability that the d_m document belongs to the topic of t_k;
- The matrix $\Phi=[\phi_{k,w}]_{K\times N}$ – reflects the probability of the word www appearing in the topic t_k.

These two matrices form the basis for clustering, comparing, visualising, and practical interpretation of the thematic structure of the corpus. For each document d_m, has:

$$\theta_m=[\theta_{m1},\theta_{m2},\ldots,\theta_{mK}], \tag{38}$$

where $\sum_{k=1}^{K}\theta_{mk}=1$.

Each element θ_{mk} – is the probability that the document d_m belongs to the topic t_k. The vector θ_m – can be considered as the thematic profile of the document in the space of dimension K. For each topic, t_k:

$$\phi_k = \left[\phi_{k1}, \phi_{k2}, \ldots, \phi_{kN}\right], \tag{39}$$

where $\sum_{w=1}^{N} \phi_{kw} = 1$.

The element ϕ_{kw} – shows how characteristic the word w is for the topic t_k. Words with the largest ϕ_{kw} – form the semantic core of the topic, that is, a set of key lexemes that best describe it.

The matrix Θ specifies a new representation of documents in the K topic space:

$$D = \{\theta_1, \theta_2, \ldots, \theta_M\}, \theta_m \in R^K. \tag{40}$$

Thus, each document is presented not through words, but through a vector of thematic weights.

For two documents d_i and d_j – described by the vectors θ_i, θ_j, their thematic similarity is calculated, for example, by the cosine measure:

$$\mathrm{Sim}\left(d_i, d_j\right) = \frac{\theta_i \cdot \theta_j}{\left\|\theta_i\right\| \left\|\theta_j\right\|}. \tag{41}$$

Alternatively, you can use the Euclidean distance:

$$\mathrm{Dist}\left(d_i, d_j\right) = \sqrt{\sum_{k=1}^{K} \left(\theta_{ik} - \theta_{jk}\right)^2}. \tag{42}$$

Small Dist values or large Sim values mean that the documents have similar thematic content. The resulting θ_m vectors can be grouped using classical clustering methods, such as K-means, Agglomerative clustering, DBSCAN or HDBSCAN. For each document, the belonging to the C_j cluster is calculated, which is formally defined as:

$$C_j = \left\{d_m / \arg\max_k \theta_{mk} = j\right\}. \tag{43}$$

Therefore, documents dominated by the same theme (maximum θ_{mk}) fall into the same cluster. For the Ukrainian News Corpus (15 topics):

Table 8

Results of the analysis of the Ukrainian news corpus (15 topics)

Document	Dominant theme	Topic Top Words	Interpretation
d_1	2	war, front, army, weapons	Military events
d_2	4	government, budget, economy, taxes	Financial policy
d_3	9	University, Education, Science, Students	Educational sphere
d_4	2	army, soldiers, front, Ukraine	Military News
d_5	9	School, Studies, Students, Teachers	Education

Thus, documents d_1 and d_4 – form one cluster (military), and d_3 and d_5 – another (educational).

It can be considered as a visualisation of the space Θ and Φ after the dimensionality is reduced. Mathematical foundations of construction:

1. Input:

- The matrix $\Theta_{M \times K}$ – is the distribution of topics in documents.
- The matrix $\Phi_{K \times N}$ – is the distribution of words in topics.

2. Dimension reduction to build a two-dimensional or three-dimensional map, the following methods are used:

- t-SNE (t-Distributed Stochastic Neighbour Embedding):

$$P_{ij} = \frac{\exp\left(-\left\|\theta_i - \theta_j\right\|^2 / 2\sigma^2\right)}{\sum_{a \neq b} \exp\left(-\left\|\theta_a - \theta_b\right\|^2 / 2\sigma^2\right)}, \tag{44}$$

where P_{ij} – is the probability of proximity of documents I and J.

- UMAP (Uniform Manifold Approximation and Projection) – building a map based on topological proximity and local data density.

3. Visualization:

- Each point is a document, its coordinates $\left(x_m, y_m\right)$ – are obtained with a decrease in dimension θ_m.
- The colour of the dot corresponds to the dominant theme.
- The distances between the points reflect thematic proximity.

To construct a semantic map of topics, the matrix Φ – is used: t_k topics with close distributions ϕ_k – are located side by side.

$$\mathrm{Dist}\left(t_i, t_j\right) = 1 - \frac{\phi_i \cdot \phi_j}{\left\|\phi_i\right\| \left\|\phi_j\right\|}. \tag{45}$$

It allows you to identify semantically related topics, for example, *education* ↔ *science*, army ↔ *politician*, and e*conomics* ↔ *business*.

Based on Θ, the model can automatically classify new texts:

$$\hat{k} = \arg\max_k P\left(t_k \middle| d_{new}\right) = \arg\max_k \theta_{new,k}. \tag{46}$$

It allows you to create automatic thematic distribution systems (for example, classification of news, scientific articles or user reviews). In the problem of searching documents for the query q:

4. The query vector θ_q – is defined as the mean of the ϕ vectors of its words:

$$\theta_q = \frac{1}{|q|} \sum_{w_i \in q} \phi_{:,w_i}. \tag{47}$$

5. The similarity between θ_q and each θ_m – is then calculated:

$$\mathrm{Sim}(q, d_m) = \frac{\theta_q \cdot \theta_m}{\|\theta_q\| \|\theta_m\|}. \quad (48)$$

It allows you to implement a semantic search that understands the content of the query, and not just the coincidence of words. Using document timestamps, you can track changes in the frequency of topics:

$$f_k(t) = \frac{1}{|D_t|} \sum_{d_m \in D_t} \theta_{m,k}. \quad (49)$$

It allows you to create dynamic thematic maps and determine which topics are gaining or losing relevance. The θ_m vectors can be used as features for further models (e.g., in recommendation, user classification, or news systems). For a corpus of Ukrainian publications with a volume of 10,000 documents, the LDA model with $K = 15$ topics gave the following results:

Table 9
Results of the analysis of the corpus of Ukrainian publications with a volume of 10,000 documents

N	Keywords (top 5)	Semantic category	Middle fraction $\overline{\theta_k}$
1	army, war, front, weapons, Ukraine	Military themes	0.18
2	government, taxes, economy, budget, hryvnia	Economics	0.15
3	education, students, university, science, school	Education and science	0.12
4	Sports, Match, Team, Championship, Players	Sport	0.09
5	culture, theatre, music, film, festival	Culture	0.07

On the semantic map, these topics have formed separate clusters, and education-science and culture-art are located close to each other, which corresponds to their real content affinity.

Thus, the estimated parameters Θ and Φ have a double value:

- mathematical – as parameters of a probabilistic model;
- semantic – as vectors of semantic representation of texts and topics.

Their use allows to form thematic clusters; build semantic maps; search and analyse information at the content level; to identify trends and regularities in large corpora of Ukrainian texts.

In the conducted study, a probabilistic model of thematic modelling of texts based on the Latent Dirichlet Allocation (LDA) algorithm for Ukrainian-language corpora of documents is built and mathematically substantiated.

1. According to the LDA model, the parameters $\Theta = [\theta_{m,k}]$ and $\Phi = [\phi_{k,w}]$ – are realisations of random variables distributed according to Dirichlet:

$$\theta_m \sim \mathrm{Dir}(\alpha), \phi_k \sim \mathrm{Dir}(\beta). \quad (50)$$

2. The full probability of the corpus of documents is derived in the form of:

$$P(W|\alpha,\beta)=\prod_{m=1}^{M}\int P(\theta_m|\alpha)\left(\prod_{n=1}^{N_m}\sum_{z_{mn}}P(z_{mn}|\theta_m)P(w_{mn}|z_{mn},\beta)\right)d\theta_m, \quad (51)$$

which specifies the basic plausibility function for LDA.

3. The use of approximation methods for estimating parameters – the variational Bayesian approach and Gibbs sampling, which provide a practically feasible assessment of the a posteriori distribution, is substantiated $P(\Theta,\Phi,Z|W,\alpha,\beta)$.

4. Analytical expressions of parameter estimates are obtained:

$$\widehat{\theta_{m,k}}=\frac{n_{m,k}+\alpha_k}{\sum_{k'}\left(n_{m,k'}+\alpha_{k'}\right)},\ \widehat{\phi_{k,w}}=\frac{n_{k,w}+\beta_w}{\sum_{w'}\left(n_{k,w'}+\beta_{w'}\right)}, \quad (52)$$

which allows for the recovery of the thematic structure of the enclosure based on the observed frequencies.

Quality characteristics of the model:

1. To assess the quality of the thematic model, the following metrics were used:

- Perplexity is a probabilistic indicator of the consistency of the model with test data.
- Coherence (UMass) – semantic consistency of topic keywords.
- NPMI (Normalised Pointwise Mutual Information) is a normalised measure of semantic proximity between topic words.

2. Empirical results for the Ukrainian corps have been obtained:

- Minimal perplexity is observed at the number of topics $K=15-25$;
- Average coherence value $C_{UMass}\approx 0.6$;
- Average $C_{NPMI}^{model}\approx 0.48$, which indicates high consistency and stability of topics.

3. It has been proven that for Ukrainian-language texts, the quality of modelling significantly depends on the preliminary processing of data – lemmatisation, clearing of stop words, and normalisation of morphological forms.

Practical significance and application:

1. Document clustering. Using the matrix Θ, documents are grouped in the thematic space:

$$C_j=\{d_m / \arg\max_k \theta_{mk}=j\}. \quad (53)$$

It allows you to form meaningful thematic clusters (for example, war, economy, education, sports, culture).

2. Construction of semantic maps. On the basis of the parameters Θ and Φ, two-dimensional mappings (*using the t-SNE* and *UMAP* methods) have been constructed, which reflect the relationships between documents and topics ϕ_k.

3. Semantic search and recommendation systems. The vectors θ_m – are used as features for comparing the content of texts:

$$\mathrm{Sim}(q, d_m) = \frac{\theta_q \cdot \theta_m}{\|\theta_q\| \|\theta_m\|}, \tag{54}$$

which allows you to implement a search by content and thematic recommendation of materials.

4. Analysis of thematic dynamics. The function of changing the popularity of topics over time has been built:

$$f_k(t) = \frac{1}{|D_t|} \sum_{d_m \in D_t} \theta_{m,k}, \tag{55}$$

which makes it possible to investigate the development of social trends, changes in media narratives and the evolution of discourses. The study showed that the LDA algorithm is an effective tool for automatic detection of the content structures of Ukrainian-language texts.

- analysis of large corpora of publications and news;
- classification of scientific articles and social messages;
- building thematic user profiles;
- Content analytics in information and educational systems.

Mathematically, the LDA model implements an optimal balance between probabilistic consistency (perplexity) and semantic meaningfulness (coherence, NPMI). The results obtained confirm that even for the morphologically complex Ukrainian language, the model provides stable, interpretable and practical results from the point of view of content analysis. Prospects for further research:

1. Improving the model by taking into account contextual relationships between words – in particular, through the use of BERTopic, Top2Vec or Contextualised Topic Models (CTM).
2. Integration of thematic modelling with deep learning methods (e.g. BERT, mBERT, XLM-R).
3. Development of Ukrainian-language corpora for teaching thematic models from various subject areas.
4. Automated comparison of thematic structures between languages for cross-cultural analysis of texts.

Thus, the mathematical model of thematic modelling, implemented on the basis of *Latent Dirichlet Allocation*, has confirmed its effectiveness for processing Ukrainian-language texts, providing not only formal grouping of documents, but also a deep semantic interpretation of the content, which opens up vast opportunities for application in scientific, educational, information and analytical systems.

In today's information environment, the amount of text data generated daily is constantly increasing. Texts are generated in the form of news messages, publications in social networks, user reviews, and scientific and business documents that require in-depth analysis. Since most of this information is unstructured, its manual processing is ineffective or even impossible with large volumes. Within the framework of this work, a system of thematic modelling of texts is being developed, which is able to automatically determine the topics present in a large corpus of documents. Such a system analyses the frequency of word use in different texts, identifies hidden patterns and groups words into topics that reflect the semantic essence of the content. Mark-up of documents or predefined categories, which significantly simplifies the analysis and allows the model to be used in a wide range of tasks. The successful implementation of such a system largely depends on the correct choice of text processing methods and thematic analysis algorithms. The accuracy of the results, the possibility of scaling, the interpretation of the output and the

adaptation of the system to different languages or types of texts depend on this. Justification of the choice of mathematical methods, models of knowledge representation, logical inference mechanisms and software tools that ensure the effective functioning of the decision-making system in the problem of thematic modelling of texts.

For practical thematic analysis of textual information, it is necessary to present unstructured text data in a formalised form suitable for machine processing. Within the framework of the developed system of thematic modelling of texts, a classic and widely used approach to the presentation of knowledge has been chosen - a vector model of the text, which reflects the frequency of use of terms in the text. The most common ways to construct a vector space are the Bag-of-Words (BoW) model and the TF-IDF (Term Frequency – Inverse Document Frequency) model. Both approaches allow you to convert text into a numerical vector, where each coordinate corresponds to a specific word from the general dictionary, and the value corresponds to the frequency of its occurrence in the document or in a unique context. The bag-of-words model is simple and effective: it counts the number of times each word from the dictionary occurs in a document and does not take into account word order or grammatical relationships. This approach allows you to quickly form a vector representation of the text and provides a sufficient quality of analysis for the task of thematic modelling. In turn, the TF-IDF model complements the BoW, taking into account not only the frequency of a word in a particular document, but also how rare or characteristic it is in the entire corpus. It allows you to reduce the impact of commonly used words (even if they are not stop words) and increase the significance of terms that are specific to a particular topic. As a result of processing the corpus of documents, a "document-term" matrix is formed, which is the basic source of knowledge for building a thematic model. Thus, the representation of knowledge in the form of a vector space is a reliable and mathematically grounded way for the further application of thematic analysis algorithms, in particular LDA.

In the developed system of thematic analysis, probabilistic thematic modelling is implemented using the Latent Dirichlet Allocation (LDA) algorithm, which is the primary mechanism of logical inference. This is one of the most well-known and proven approaches in the problems of analysis of unstructured texts, which makes it possible to identify hidden (latent) thematic structure in large corpora of documents. The LDA algorithm is based on the idea that each document is The model assumes that the process of writing each document took place with the participation of several topics, which in different proportions "influenced" the choice of words in the text. . This approach is much closer to the real functioning of the language, when the text often combines several semantic directions. The process of logical inference consists in building a probabilistic model of the distribution of words by topics and topics in documents based on input texts presented in the form of vectors (for example, through Bag-of-Words or TF-IDF). The algorithm uses a Bayesian approach and includes two hyperparameters – alpha (α) and beta (β), which control the "blurring" of topics on documents and words on topics, respectively. The result of logical inference is two main components:

- θ (theta) – distribution of topics in each document;
- φ (phi) – distribution of words in each topic.

Thus, the model makes decisions about the thematic structure of texts not on the basis of external markers or keywords, but on the basis of statistical patterns of word co-use. It allows for the detection of topics even when they are not explicitly labelled in the text, which is critical for the analysis of large unstructured data sets. The selected LDA model provides high interpretation, which makes it convenient not only for internal machine analysis but also for further presentation of the results to the end user (analyst, researcher, journalist, etc.). Within the framework of the project on thematic modelling of texts, the Latent Dirichlet Allocation (LDA) method was chosen as the main algorithm for identifying topics in the corpus of documents. This choice is justified both from a theoretical and practical point of view, since LDA combines ease of implementation, high interpretability of results, and adaptability to different types of texts. First of all, LDA is a method

of unsupervised learning. It means that to build a model, you do not need pre-marked data – documents with specified topics. The model independently detects the structure of the input data and identifies hidden topics, which is especially valuable in conditions of limited or absent resources for manual labelling. It also allows the algorithm to be used in problems where the subject matter of the documents is unknown or unpredictable. Another significant advantage is the high interpretation of the results. In LDA, each topic is presented as a distribution of words, which allows it to be formulated as a set of key terms. Similarly, each document is described as a combination of topics with appropriate weights. This structure enables the analyst not only to see which topics exist in the corpus, but also to understand why a particular text belongs to a specific topic, exploratory analysis or decision-making. In addition, LDA is a flexible tool, as it allows you to change the number of topics depending on the user's needs or the size of the case. In the process of experiments, it is possible to vary the parameters of the model (the number of topics, the number of words in the topic, word frequency thresholds, etc.), choosing the optimal representation of the content of texts. Explainability and customizability, the LDA algorithm occupies a central place among the classical methods of thematic analysis of texts. Its effectiveness and reliability have been proven in numerous applied projects – from the analysis of customer reviews to the study of scientific publications, which confirms the feasibility of its choice for the implementation of the system being developed. In the field of thematic analysis of texts, there are a large number of approaches that allow you to group documents by content or identify the main topics. However, each of these methods has its own characteristics, limitations and suitability for certain types of tasks. Within the framework of this work, a comparative analysis of the most common approaches was carried out: Latent Dirichlet Allocation (LDA), K-means, Naive Bayes/SVM, BERTopic and NMF. The primary purpose of this analysis is to substantiate the feasibility of using LDA as the core of the system being developed. The main criteria for comparison:

- Availability of training tags (whether the method of manual marking of documents is required);
- Interpretation of the results (whether it is clear why the document is related to a specific topic);
- Taking into account the context (whether the method takes into account the meaning of words depending on neighbouring words);
- Performance/resource intensity (does it require a lot of computing resources);
- Flexibility (whether it is possible to change the number of themes, parameters, and scales).

Table 10
Comparative methods

Method	Needs labels	Interpretation of the result	Taking into account the context	Speed of work	Resource intensity	Flexibility
LDA	No	High	N/a	Average	Low	High
K-means	No	Low	N/a	High	Low	Limited
NB	Yes	High	N/a	High	Low	Low
BERTopic	No	Average	Is	Low	High	High
NMF	No	Average	N/a	High	Low	Limited

Latent Dirichlet Allocation (LDA) is a probabilistic model that allows you to automatically detect topics in texts without the need for training labels. Each topic is presented as a list of keywords, and each document is described as a set of topics with different weights. It allows the user not only to see what topics are present in the corpus, but also to understand which words characterise them. Although LDA does not take into account context (i.e., all words are considered independently), it remains one of the best models for its combination of simplicity, explainability, and efficiency.

The K-means clustering method is often used to group similar documents based on a vector representation (for example, TF-IDF). Although it allows you to automatically form groups of documents, it does not describe the topic – there is no list of words that characterise each cluster. Thus, it is more suitable for dividing into "similar texts" than for in-depth semantic analysis. Also, the number of clusters is set manually and needs to be configured.

Naive Bayes / SVM – these models belong to supervised learning and are used for classical classification of texts. They show promising results, but require a study sample with topics already marked. It makes it impossible to use in problems where the topic is not known in advance. In the context of our system, which is supposed to work without labels, these approaches are not suitable.

BERTopic is a modern model that combines the capabilities of transformers (for example, BERT) with clustering. It takes into account the context of the word in the sentence, which makes it possible to highlight topics much more accurately. But its main drawback is high computing requirements (a GPU is required, a lot of memory). In addition, its results are less interpretable for the non-professional user. For tasks where simplicity and transparency are essential, BERTopic is less convenient.

NMF (Non-negative Matrix Factorisation) is a matrix decomposition method that is also used for thematic analysis. It allows you to find topics as factors in the decomposition of the document-term matrix. Although NMF performs well on small cases, it is less stable and less explainable than LDA. Also, it does not give a clear probabilistic representation of topics in documents.

For the implementation of the system of thematic modelling of texts, a set of software tools was chosen that provides effective processing, modelling and visualisation of results. All tools are open, free, widely supported by the community, and have documentation that makes them easy to implement and modify. Python was chosen as the primary development language due to its popularity in the field of natural language processing (NLP), the availability of numerous libraries, convenient syntax, and active support.

Python acts as the system's main implementation environment. This programming language has a rich set of tools for scientific computing, data analysis, and machine learning. It allows you to effectively work with texts, process, visualise, and integrate with web interfaces or graphical shells. Python must support the processing of Ukrainian-language texts, which is relevant for our system.

The Gensim library, which is one of the world's most popular LDA implementations, was chosen to build the thematic model. It allows you to work effectively with large volumes of text, implements both model training and topic revision, saves models, works with dictionaries, etc. Its key advantages are its high performance and ease of integration with other Python modules.

The pyLDAvis library is used to visualise the results of the thematic modelling. It allows you to present the topics you find in the form of an interactive map, which shows the distribution of topics, the weight of keywords, and their connections. Such visualisation makes it much easier for the user to interpret the results, especially in cases of a large number of topics or documents.

The Pandas library is used to store and process modelling results – tables with the distribution of topics across documents, frequency tables, export to CSV format, etc. It is also used in the stage of pre-processing the corpus of texts and aggregation of data for analysis.

At the stage of text pre-processing, NLTK (Natural Language Toolkit) or spaCy libraries are used. They provide stop word removal, tokenisation, lemmatisation, and other linguistic operations. NLTK has built-in dictionaries of Ukrainian stop words, and spaCy can be extended to work with the Ukrainian language through third-party modules (spacy-uk).

All of these tools are open-source, have an active community, and are constantly updated. They allow you to build robust, flexible, and scalable case study systems that can be easily adapted to different languages and types of text data. In the process of choosing technologies for the implementation of the system of thematic modelling of texts, a number of alternative tools and platforms with similar functionality were also considered. These include cloud API services (Google Cloud Natural Language API, IBM Watson Natural Language Understanding), platforms with ready-made analysis templates such as MonkeyLearn, and modern transformer-based models – in particular, BERTopic. Despite their potential advantages, they turned out to be less suitable for the tasks of this project, which will be justified below. Google Cloud Natural Language API – this service offers ready-made functionality for classifying texts, analysing sentiment, extracting entities, etc. However, in the context of thematic modelling, it has a number of limitations:

- Thematic categories are pre-fixed; they cannot be changed or tailored to body.
- Support for the Ukrainian language is limited or non-existent.
- The service operates on a paid model, which creates dependence on the budget and external infrastructure.
- The lack of control over internal modelling processes makes it impossible to explain the results (i.e., interpretation decreases).

IBM Watson Natural Language Understanding also offers a high-level platform for analysing texts, including classification, emotional analysis, and key concepts. However:

- The platform is focused on English-speaking and global markets, but the Ukrainian language is not supported.
- Topics are defined based on built-in categories, which are not suitable for tasks where you need to highlight new or specialised issues.
- The Watson interface is built like SaaS, which means limited customisation and integration into on-premises solutions.
- Like Google NLP, Watson is a commercial closed product, and therefore not suitable for fully open research or academic solutions.

MonkeyLearn is a no-code/low-code text analysis platform that allows you to create your own classifiers and topic analysers using templates. Although it is user-friendly without technical training, its capabilities have a number of significant limitations:

- The work is mainly based on classification by labels, and not on full-fledged thematic modelling.
- All computing takes place in cloud, and therefore, local integration is not possible.
- Limited support for the Ukrainian language.
- The free version has very tight volume restrictions.

BERTopic is a modern Python library that combines the capabilities of transformers (BERT) with clustering. It is able to take into account the context of words and form dynamic, flexible topics. However, its use in the project was rejected due to:

- Extremely high resource consumption – you need a powerful GPU, a significant amount of memory, and a complex environment setup.
- Low interpretation: topics are defined through vectors rather than transparent word frequencies, so it is much more difficult to explain why a document belongs to a topic.
- Support for the Ukrainian language is partial and requires downloading separate models.

As a result of the analysis, it was found that for tasks related to the detection of topics in unstructured text data, the optimal solution is to use Latent Dirichlet Allocation (LDA) – a probabilistic model of thematic modelling that does not require training labels and allows you to find hidden topics in documents based on the distribution of words. LDA provides a high level of interpretation of results, which is essential for making informed decisions and creating reports. In addition, the model is flexible, highly scalable, and does not require excessive computing resources. The representation of knowledge in the system is carried out through vector representations of the text - Bag-of-Words and TF-IDF models. They allow you to turn natural language into structures that are convenient for processing by machine learning algorithms. It ensures effective pre-processing of information before transferring it to the thematic model. In the process of logical inference, the model makes decisions based on the distribution of probabilities: it determines the probability with which certain words belong to a particular topic, as well as in what ratio the topics are present in each document. This approach imitates the real process of reading and analysing the text, allowing the system to reveal the semantic structure without human intervention. A comparative analysis with alternative methods – K-means, Naive Bayes, BERTopic, NMF, as well as with external platforms such as Google NLP API, IBM Watson and MonkeyLearn – showed that although these tools have their own advantages, none of them combine simplicity, flexibility, openness, interpretation and local support as LDA does in the Python environment. Separately, the choice of software tools such as Python, Gensim, pyLDAvis, Pandas, NLTK and spaCy was justified. All of them are open-source, well-documented, support Ukrainian language processing, and ensure efficient implementation of both the main computing and visualisation components of the system. Thus, the work made it possible to lay a methodological basis for the practical implementation of the program module of thematic analysis of texts. Carefully selected methods and tools ensure the accuracy, adaptability, and efficiency of the system being developed and that it is ready for further software implementation within the following stages of work.

5. Experiments

Now we are in a modern digital environment, and the volume of textual information is constantly growing. Every day, users, government agencies, media, scientific organisations, and other sources generate vast amounts of text data, including news, reports, publications, social media posts, and more. In the conditions of such an information flow, there is a need for a fast, effective and automated analysis of the content of texts. Thematic modelling is an essential tool in Natural Language Processing (NLP) that allows you to discover hidden structures in a collection of documents. It will enable you to determine what is being said in large arrays of texts, without the need for their manual reading, which is especially important for media analytics, public opinion research, information monitoring, library systems, etc. This topic is especially relevant for Ukrainian-language sources, where resources for deep language analysis are still limited, and ready-made analysis systems often do not support the Ukrainian language at the proper level.

Table 11
Purpose of the study

#	Purpose
1	Development of software for thematic modelling of Ukrainian-language texts.
2	Automatic classification of documents by topic.
3	Identifying the keywords that shape each topic.
4	Determining the topic of the new text.

5	Evaluating the quality of a thematic model using coherence.
6	Visualisation of the results of thematic modelling.

The key problem is information overload: an analyst or user is not able to manually process and summarise dozens or hundreds of documents, especially when they are unstructured. Due to the lack of tools for automated thematic analysis, time is lost, the likelihood of errors increases, and data-driven decision-making is also tricky.

Table 12
Applied methods

#	Method/Tool	Appointment
1	LDA	Thematic modelling, identification of probabilistic distributions of topics
2	NLP	Pre-processing of texts: tokenisation, clean-up, normalisation
3	Stanza	Lemmatisation of Ukrainian words based on deep learning
4	Gensim	Creating an LDA model, dictionary, and corpus, and calculating coherence
5	pyLDAvis	Interactive visualisation of topics and their structure

Table 13
Software architecture and structure

#	Module Name	Appointment
1	Data processing module (pre-processing)	Performs text cleanup, tokenisation, lowercase, stop word removal, and lemmatisation (via Stanza).
2	Thematic Modelling Module (LDA)	Trains the LDA model based on the corpus of texts created after pre-processing; forms topics as a distribution of words.
3	Coherence Assessment Module	Calculates the topic coherence metric (c_v) using the CoherenceModel from the Gensim library.
4	Interpreting and generating topic names	Generates conditional topic names based on keywords; allows you to better interpret the semantics of topics.
5	Results visualisation module	Outputs an interactive topic map (pyLDAvis) and a coherence graph for a different number of topics.
6	New Texts Classification Module	Accepts new text, performs processing and classifies, determining the probabilities of belonging to topics.

The software for thematic modelling of texts is implemented in the form of a modular system, where each component is responsible for a separate functionality. This approach provides flexibility, code reuse, and ease of scaling or adapting to other languages or tasks.

```
stop_words = {
    'і', 'в', 'во', 'не', 'що', 'він', 'на', 'який', 'з', 'у', 'та', 'до',
    'це', 'по', 'вона', 'його', 'її', 'ми', 'вони', 'ви', 'ти', 'ж', 'як',
    'так', 'а', 'але', 'чи', 'таким', 'того', 'було', 'була', 'тому', 'такий',
    'є', 'був', 'буде', 'може', 'ні', 'де', 'кого', 'кого-небудь', 'ніхто'
}
```

```
def preprocess(text):
    doc = nlp(text)
    lemmas = []
    for sentence in doc.sentences:
        for word in sentence.words:
            lemma = word.lemma.lower()
            if lemma.isalpha() and lemma not in stop_words and len(lemma) > 2:
                lemmas.append(lemma)
    return lemmas
```

```
# 1. Завантаження датасету
df = pd.read_csv("president_newss.csv")
```

```
# 2. Перевірка структури
print(df.columns)  # переконайся, що є колонка 'text'
```

```
Index(['title', 'text'], dtype='object')
```

```
df.head()
```

```
import warnings
from tqdm import tqdm

warnings.filterwarnings("ignore", category=DeprecationWarning)

texts = df['text'].dropna().tolist()
processed_texts = [preprocess(text) for text in tqdm(texts)]
```

Figure 3: Data processing module (pre-processing)

```
lda_model = LdaModel(
    corpus=corpus,
    id2word=dictionary,
    num_topics=22,          #кількість тем
    random_state=42,
    passes=10,
    alpha='auto'
)

winsound.Beep(1000, 1000)
```

```
# 🔹 Виведемо теми
topics = lda_model.print_topics(num_words=7)
for i, topic in topics:
    print(f"Тема {i + 1}: {topic}")
```

Figure 4: Thematic Modelling Module (LDA)

```python
def compute_coherence_values(dictionary, corpus, texts, start, limit, step):
    coherence_scores = []
    models = []
    for num_topics in range(start, limit + 1, step):
        model = LdaModel(
            corpus=corpus,
            id2word=dictionary,
            num_topics=num_topics,
            random_state=42,
            passes=10,
            alpha='auto'
        )
        coherence_model = CoherenceModel(
            model=model,
            texts=texts,
            dictionary=dictionary,
            coherence='c_v'
        )
        score = coherence_model.get_coherence()
        coherence_scores.append(score)
        models.append(model)
        print(f"num_topics={num_topics} → когерентність={score:.4f}")
    return models, coherence_scores

winsound.Beep(1000, 1000)
```

```python
# Налаштування діапазону
start = 12
limit = 40
step = 1
```

```
num_topics=12 → когерентність=0.5135
num_topics=13 → когерентність=0.4749
num_topics=14 → когерентність=0.4813
num_topics=15 → когерентність=0.4880
num_topics=16 → когерентність=0.4732
num_topics=17 → когерентність=0.4655
num_topics=18 → когерентність=0.4743
num_topics=19 → когерентність=0.4698
num_topics=20 → когерентність=0.4582
num_topics=21 → когерентність=0.4733
num_topics=22 → когерентність=0.5167
num_topics=23 → когерентність=0.4711
num_topics=24 → когерентність=0.5001
num_topics=25 → когерентність=0.4782
num_topics=26 → когерентність=0.4884
num_topics=27 → когерентність=0.5102
num_topics=28 → когерентність=0.5025
num_topics=29 → когерентність=0.4757
num_topics=30 → когерентність=0.4741
num_topics=31 → когерентність=0.4414
num_topics=32 → когерентність=0.4616
num_topics=33 → когерентність=0.4673
num_topics=34 → когерентність=0.4808
num_topics=35 → когерентність=0.4744
num_topics=36 → когерентність=0.4622
num_topics=37 → когерентність=0.4611
num_topics=38 → когерентність=0.4547
num_topics=39 → когерентність=0.4545
num_topics=40 → когерентність=0.4732
```

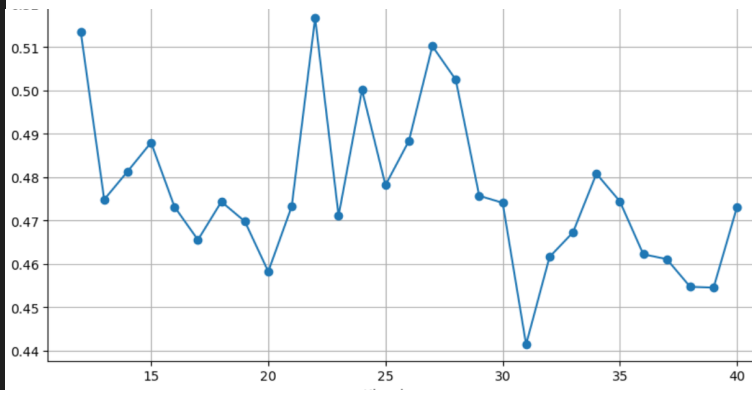

Figure 5: Coherence Assessment Module

```
Тема 21: 0.016*"римський" + 0.014*"xiv" + 0.013*"папа" + 0.010*"святий" + 0.010*"лев" + 0.009*"мес" + 0.007*"петро"
Тема 13: 0.056*"індія" + 0.014*"індійський" + 0.014*"міністр" + 0.013*"між" + 0.013*"україна" + 0.013*"модя" + 0.010*"культура"
Тема 3: 0.012*"комісар" + 0.010*"оон" + 0.009*"гранді" + 0.009*"філіппо" + 0.009*"верховний" + 0.006*"біженець" + 0.005*"українець"
Тема 18: 0.015*"фінляндія" + 0.010*"обсе" + 0.006*"словацький" + 0.005*"александр" + 0.005*"для" + 0.005*"михал" + 0.005*"валтонен"
Тема 16: 0.030*"рятувальний" + 0.027*"начальник" + 0.027*"державний" + 0.022*"пожежній" + 0.021*"частина" + 0.019*"рятувальник" + 0.019*"дснс"
Тема 6: 0.012*"свій" + 0.011*"народ" + 0.011*"кримськотатарський" + 0.009*"геноцид" + 0.008*"україна" + 0.008*"грузія" + 0.008*"володимир"
Тема 15: 0.019*"україна" + 0.018*"компанія" + 0.014*"президент" + 0.011*"володимир" + 0.011*"зеленський" + 0.010*"також" + 0.009*"міністр"
Тема 19: 0.015*"україна" + 0.013*"зеленський" + 0.013*"бути" + 0.011*"володимир" + 0.011*"президент" + 0.010*"український" + 0.010*"олена"
Тема 4: 0.017*"під" + 0.012*"окупант" + 0.012*"знищити" + 0.011*"час" + 0.011*"рік" + 0.009*"україна" + 0.009*"позиція"
Тема 14: 0.015*"україна" + 0.012*"чехія" + 0.012*"для" + 0.009*"бути" + 0.008*"ветеран" + 0.008*"також" + 0.007*"володимир"
Тема 17: 0.025*"україна" + 0.018*"президент" + 0.014*"держава" + 0.013*"зеленський" + 0.013*"володимир" + 0.011*"український" + 0.010*"про"
Тема 10: 0.044*"україна" + 0.013*"підтримка" + 0.012*"угода" + 0.012*"сфера" + 0.011*"сторона" + 0.011*"безпека" + 0.010*"для"
Тема 1: 0.014*"україна" + 0.011*"під" + 0.011*"окупант" + 0.011*"російський" + 0.009*"бригада" + 0.009*"воїн" + 0.009*"президент"
Тема 22: 0.022*"україна" + 0.018*"зеленський" + 0.018*"володимир" + 0.016*"президент" + 0.013*"держава" + 0.012*"наш" + 0.008*"глава"
Тема 20: 0.030*"україна" + 0.025*"президент" + 0.017*"володимир" + 0.016*"зеленський" + 0.015*"безпека" + 0.014*"бути" + 0.013*"для"
Тема 7: 0.030*"україна" + 0.015*"президент" + 0.013*"зеленський" + 0.013*"володимир" + 0.011*"український" + 0.011*"цей" + 0.010*"наш"
Тема 9: 0.022*"україна" + 0.015*"для" + 0.014*"бути" + 0.010*"держава" + 0.010*"президент" + 0.009*"цей" + 0.009*"також"
Тема 12: 0.021*"україна" + 0.017*"бути" + 0.012*"президент" + 0.012*"володимир" + 0.012*"для" + 0.012*"зеленський" + 0.011*"мир"
Тема 2: 0.032*"україна" + 0.022*"президент" + 0.017*"володимир" + 0.017*"зеленський" + 0.015*"держава" + 0.013*"про" + 0.011*"для"
Тема 11: 0.040*"україна" + 0.021*"президент" + 0.020*"зеленський" + 0.020*"володимир" + 0.017*"підтримка" + 0.016*"держава" + 0.012*"також"
```

```
Тема 1: Під і окупант у контексті знищити ➤ під, окупант, знищити, україна, російський, рік
Тема 2: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, держава, про
Тема 3: Час і український у контексті фіцпатрік ➤ час, український, фіцпатрік, інструктор, браян, під
Тема 4: Під і час у контексті бригада ➤ під, час, бригада, україна, російський, група
Тема 5: Україна і норвегія у контексті зеленський ➤ україна, норвегія, зеленський, володимир, мелон, президент
Тема 6: Школа і президент у контексті україна ➤ школа, президент, україна, дитина, штат, володимир
Тема 7: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, український, цей
Тема 8: Україна і ступінь у контексті орден ➤ україна, ступінь, орден, бути, президент, iii
Тема 9: Україна і для у контексті бути ➤ україна, для, бути, президент, держава, зеленський
Тема 10: Україна і сторона у контексті литва ➤ україна, сторона, литва, сполучений, сфера, цей
Тема 11: Україна і президент у контексті зеленський ➤ україна, президент, зеленський, володимир, підтримка, держава
Тема 12: Україна і бути у контексті президент ➤ україна, бути, президент, володимир, зеленський, мир
Тема 13: Нідерланд і каспар у контексті кунінор ➤ нідерланд, каспар, кунінор, японія, мацуда, велдкамп
Тема 14: Людина і удар у контексті україна ➤ людина, удар, україна, начальник, бути, володимир
Тема 15: Бути і виробництво у контексті україна ➤ бути, виробництво, україна, оборонний, компанія, президент
Тема 16: Державний і рятувальний у контексті начальник ➤ державний, рятувальний, начальник, пожежній, частина, область
Тема 17: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, німеччина, держава
Тема 18: Обсе і для у контексті словацький ➤ обсе, для, словацький, юти, фінляндія, валтонен
Тема 19: Україна і бути у контексті рабин ➤ україна, бути, рабин, український, президент, зеленський
Тема 20: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, мир, безпека
Тема 21: Римський і ватикан у контексті святий ➤ римський, ватикан, святий, папа, престол, xiv
Тема 22: Україна і зеленський у контексті володимир ➤ україна, зеленський, володимир, президент, держава, наш
```

Figure 6: Theme interpretation and name generation module

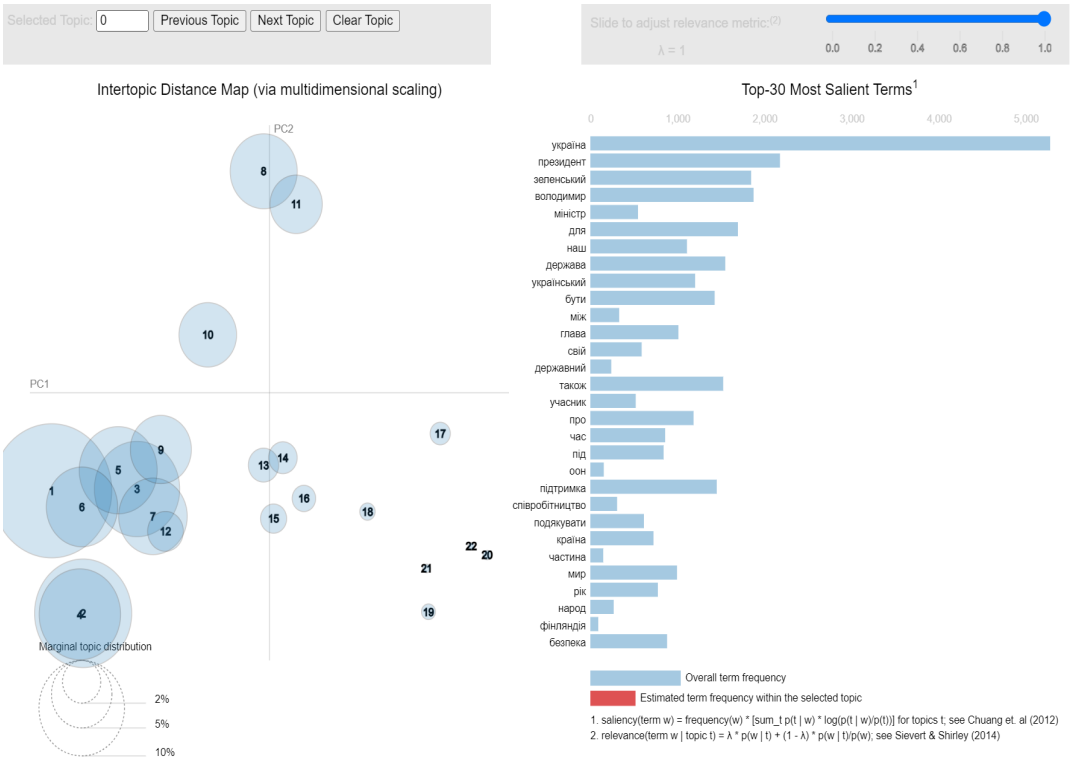

Figure 7: Results visualisation module

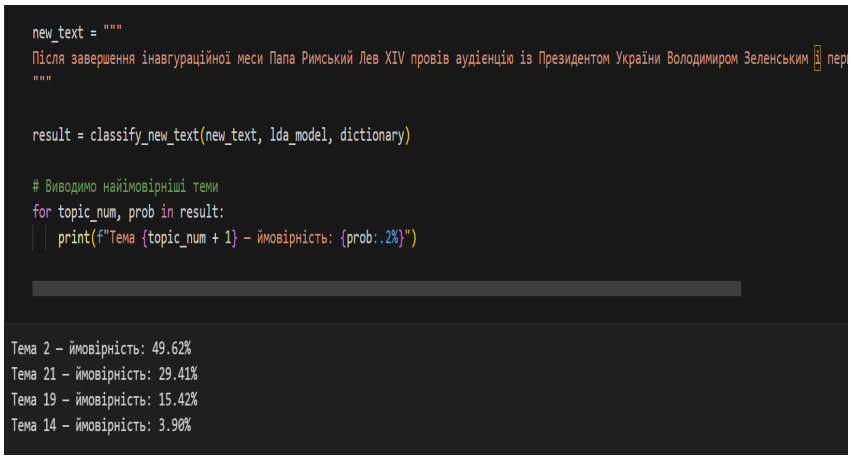

Figure 8: New Texts Classification Module

The entire software architecture is built on the principles of separation of responsibilities and extensibility. Each module performs a distinct logical function and interacts with others through well-defined interfaces. It makes it easy to maintain, modify, or integrate the system into more complex information systems. The knowledge base in the developed software is presented not in the form of traditional logical rules, but in the form of structured data and interpretation rules that allow the system to make decisions about the thematic affiliation of the text. It is formed as a result of training the LDA model and is used to automatically summarise the content.

Table 14Database components

№	Component	Description
1	Glossary of terms	A unique set of all words (tokens) included in the corpus of texts after pre-processing. Created using gensim.corpora.Dictionary.
2	Topic clusters (theme model)	The result of the LDA model is that each topic is a cluster of the most likely words. For example: topic 1 = ['president', 'office', 'council', 'government'].
3	Topic Keywords	Defined as the top-N terms with the highest probability for each topic. These words form the semantic core of the topic.
4	Topic Name Generation Rules	The generate_smart_title(keywords) function applies simple semantic rules to assign a human-oriented name to a topic. For example, if 'president' and 'office' are among the keywords, then the topic name = 'Presidential activities'.

Example of a topic structure
{ "topic_id": 7,
"keywords": ["child", "school", "education", "student", "support"],
"title": "Education and Children"}

```
Тема 21: 0.016*"римський" + 0.014*"xiv" + 0.013*"папа" + 0.010*"святий" + 0.010*"лев" + 0.009*"мес" + 0.007*"петро"
```

In software, logical inference is not implemented through traditional if–then, but through statistical machine learning algorithms, in particular LDA thematic modelling. Inference is carried out through probability distributions and the calculation of the similarity of topics.

Table 15
Logical inference mechanisms

№	Component	Description
1	LDA model	Construction of a statistical model that forms probabilistic distributions of words in topics and topics in documents. Each text is presented as a vector with the weights of belonging to each subject.
2	Algorithm for classifying new text	The new text is processed (lemmatisation, tokenisation), converted to the BoW format, and passed to LDA, which returns the probability vector. The highest probability determines the topic.
3	CoherenceModel (coherence score)	Calculates the semantic consistency of topics using the c_v metric, which is based on the mutual appearance of terms in documents. It is used to evaluate the quality of the model.

Stages of logical inference in the system

- The model is trained on the basis of the processed body of documents.
- Each topic becomes a probabilistic distribution of words.
- Each document receives a distribution of topics, such as:
- Coherence determines how stable and consistent the topics are (higher the better).
- New text can be classified via lda_model.get_document_topics() - without the need to retrain it.

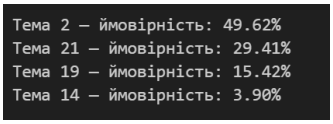

Figure 9: The result of logical inference

Thus, the system automatically concludes about the topic of the new document without manual intervention. During the operation of the software, various data structures are formed and used. They are necessary for storing the corpus of texts, a glossary of terms, the results of modelling topics and the subsequent classification of new documents. The following are the key structures that are stored or used in memory at runtime.

Table 16
Data structure

№	Structure Name	Format/Type	Appointment
1	processed_texts.pkl	.pkl file (Pickle)	Saves preprocessed texts as lists of lemmas. Allows you not to repeat preprocessing when restarting.
2	dictionary	gensim.corpora.Dictionary	A unique glossary of terms generated from processed texts. Each word has a unique ID.
3	corpus	List of lists (Bag-of-Words)	Representing each document as a dictionary-based frequency vector of words. Required for LDA training.

4	lda_model	Gensim.models.LdaModel object	A model that contains topics, their distributions, probabilities, and parameters. Retains knowledge of the model after training.
5	lda_model.print_topics()	Method	Returns a list of topics with their keywords and probabilities. It is used to interpret the results.
6	lda_model.get_document_topics()	Method	Returns a probabilistic distribution of topics for a single document. It is used to classify new texts.

The software has a text-based interface in the form of an interactive Jupyter Notebook, which provides a convenient user experience with the system. All the main functions are implemented through clear code blocks with output and visualisation.

Table 17
Interface and functionality

№	Function	Description
1	Output of topic keywords	For each generated topic, a list of keywords with the highest probabilities is displayed. It allows you to interpret the content of the topic.
2	Automatic naming of topics	Implemented through the generate_smart_title(keywords) function, which creates a conditional name of the topic based on keywords (for example, "Presidential activity", "Education and science").
3	Interactive Visualisation (pyLDAvis)	Displays topics in the form of circles, shows the relationships between them and keywords. The user can click on the issues and view their content.
4	Save models and data.	The processed data (processed_texts), dictionary, corpus, and trained model are stored in .pkl files, which avoids reprocessing and retraining.
5	Classification of new texts	The user can enter any new Ukrainian text, and the program will automatically determine which topic it most likely belongs to.
6	Inference of coherence	The system outputs an assessment of the quality of the model based on the c_v metric, which allows you to choose the optimal number of topics.
7	Graph of quality dependence on the topics	A graph is created to analyse how coherence changes with a different number of issues (e.g., 5–40). It helps to automatically select the best model.

The software's interface is built to be intuitive for both technical and non-technical users. It allows you to both study the structure of topics in the corpus and analyse new documents in real time. In the developed software, all modules function within a single data processing flow. Each component does its part of the work and passes the results to the input of the next module. This modular structure allows for consistent processing, resource overuse, and flexibility for modifications.

CSV → Word Processing → Saving → Building a Corpus → LDA Model →
→ Topic generation → Coherence → Visualization → Analysis of new textsTable 18

Co-operation processes

Stage	Process description	Participation in modules
1	Loading and pre-processing of texts	The user imports a .csv file → The data processing module cleans and lemmatises the texts.
2	Saving an intermediate result	The processed texts are stored in processed_texts.pkl for reuse.
3	Creating a dictionary/corpus	The data is passed to the dictionary and corpus via Gensim.
4	Training the LDA model	The corpus and dictionary are passed to the Thematic Modelling Module, where a theme model is created.
5	Model Quality Assessment	The model is passed to the Coherence Evaluation Module, where the c_v is calculated.
6	Visualisation of results	The theme model goes to the Visualisation Module, where an interactive theme map (pyLDAvis) is built.
7	Generating Topic Names	The keywords of each topic are analysed in the Interpretation Module, which assigns a human name.
8	Classification of new text	The user enters the text;it is processed, transmitted to lda_model → model returns the belonging probability to the topics.

Table 19
System Requirements

Category	Requirement
Operating system	Windows 10 / 11, Linux, macOS
Python	Version 3.9 or higher
Required libraries	stanza, gensim, pyLDAvis, pandas, matplotlib, pickle, tqdm
Development environment	Recommended: Jupyter Notebook / Jupyter Lab
Internet connection	Only required to load the stanza language model

Install dependencies pip install stanza gensim pyLDAvis pandas matplotlib tqdm. A parser was also developed specifically for this project, thanks to which it was possible to assemble my own unique data set from articles from the President's Office, which helped to train the model well for further use.

Table 20
Steps of use

Step	Description
1	Running pre-processing: execute the pre-process(text) function, which will clean and lemmatise texts
2	Creating a dictionary and corpus: use gensim.corpora.Dictionary and corpus = [dictionary.doc2bow(text) for text in processed_texts]
3	Model Training: Building an LDA Model via LdaModel(...)
4	Model Estimation: Coherence Calculation via CoherenceModel
5	Topic Visualisation: Use pyLDAvis to create a theme map
6	Parsing new text: call lda_model.get_document_topics() for a new document

Table 21
Description of the developed news parser from the website of the President of Ukraine

Characteristic	Description
Software Name	News parser from the website of the President of Ukraine
Appointment	Automatically collect news headlines and texts from the "Administration" section.
Link to source	https://www.president.gov.ua/news/administration
Result of work	CSV file with news headlines and texts

Table 22
Technical implementation

Component	Description
Programming language	Python 3.x
Library	selenium, bs4, csv, time
Runtime	On-premises environment (Windows with ChromeDriver installed)
Access method	Via Chrome Browser Control with Selenium

Table 23
Algorithm of work

Stage	Description

1. Initialising the driver	Running the Chrome browser in headless mode with the specified user-agent
2. Page loading	Go to the news page with the ?page=n parameter
3. Collection of news links	Search for blocks .item_stat.cat_stat, from each, the first reference is taken
4. Header Extraction	From the tag <h1 itemprop="name">
5. Text extraction	From the tag, all <p>
6. Verification	Skip news without text, prevent duplication
7. Conservation	Saving the result to a file president_news.csv

Table 24
Features of implementation

Peculiarity	Explanation
Avoiding duplicates	From each .item_stat.cat_stat block, only the first <a is taken>
Selective Content Collection	Text is taken only from articleBody, which excludes footers/menus/meta.
Dynamic content expectation	WebDriverWait is used to wait for the DOM to fully load
Work in headless mode	Can be run in the background without displaying the browser
Flexible scaling	Can be expanded to any number of pages (via the pages parameter)

When creating a parser, there was a big problem - the site president.gov.uaStructure of the output CSV file in Fig. 10. A big problem arose when creating the parser - the website president.gov.ua uses protection against automated requests (bots).This protection includes filtering requests from libraries like requests, even with fake headers (User-Agent). In particular, when using standard parsing through requests and BeautifulSoup, the server returned a 403 Forbidden response code, which indicates that the request was blocked. To circumvent this limitation, the project implemented automation of interaction with the site through the Google Chrome web browser, using the Selenium WebDriver tool and the ChromeDriver driver. It allows you to emulate the behaviour of a real user - open pages in the browser, load dynamic JavaScript content, interact with DOM elements and wait for the page to be fully rendered. The parser works in headless mode, that is, the browser does not open graphically, but all processes related to the display and processing of the web page are performed as in a real browser. It allows you to discreetly bypass anti-bot protection, while maintaining a high processing speed and minimal load on the system. Also, to minimise detection by security mechanisms, custom headers of HTTP requests were installed, including User-Agent, Accept-Language, Referer, and others, which simulate a typical request from an ordinary browser user. In addition, the code implements waiting (WebDriverWait) so that you do not try to extract information before the site fully loads the content via JavaScript. Thanks to this solution, the developed software works stably with the

official website of the President of Ukraine, bypassing server checks for the bot and ensuring the correct extraction of news texts.

	title	
1	роки для досягнення миру та можливі майданчики для розмови з росіянами	дей. За його словами, Україна готова до чесної дипломатії і необхідно забезпечити, щоб Росія
2	Президент України зустрівся з Кронпринцом Норвегії	український оборонно-промисловий комплекс. Окрема увага – посиленню культурних зв'язків
3	Володимир Зеленський провів розмову з Александром Стуббом	ві результати, мають бути жорсткі наслідки. Сьогодні в Президента України також заплановані
4	Прем'єр-міністр Австралії обговорили додатковий санкційний тиск на Росію	у участь Австралії в коаліції охочих і гарантуванні безпеки для України після досягнення справ
5	Володимир Зеленський провів зустріч із Джей Ді Венсом і Марко Рубіо	адіслав Президенту Трампу листа з новими пропозиціями щодо співпраці в оборонно-промисл
6	в аудієнцію з Президентом і першою леді України – першу для глав держав	опомагати тим, хто цього потребує. Президент і перша леді запросили Понтифіка здійснити апо
7	ки і перша леді взяли участь в інавгураційній месі Папи Римського Лева XIV	ного конклаву після смерті Папи Франциска. Роберт Френсіс Превост став 267-м Папою Римс
8	ворив із Президенткою Швейцарії наступні дипломатичні кроки заради миру	Окрему увагу приділили продовженню допомоги Україні, зокрема фінансуванню участі швейцар
9	остороння співпраця: Володимир Зеленський провів зустріч із Марком Карні	а Карні приїхати до України з візитом. Марк Карні підтвердив запрошення Главі держави взяти
10	Президент України провів зустріч із Президентом Чорногорії	ому процесі. Крім того, президенти узгодили графік найближчих заходів за участю лідерів краї
11	ленський і Дік Схооф обговорили продовження військової підтримки України	Україні. Лідери обговорили потреби України, зокрема зміцнення ППО та прямі інвестиції в укра
12	нський обговорив із Метте Фредеріксен дипломатичні зусилля заради миру	ння від головування Данії в Раді ЄС, яке розпочнеться 1 липня. Україна розраховує на підтрим
13	іх відносин: Президент України провів зустріч із Прем'єр-міністром Болгарії	і для всього Чорноморського регіону. Президент України та Прем'єр-міністр Болгарії приділили
14	идент України зустрівся з Прем'єр-міністром Швеції Ульфом Крістерссоном	Росію. Окрема увага – співпраці в межах коаліції охочих, напрацюванню гарантій безпеки та о
15	Володимир Зеленський зустрівся з президентами Євроради та Єврокомісії	ідкриття кластерів найближчим часом, а також торговельно-економічні відносини між Україною
16	восторонніх відносин: Володимир Зеленський провів зустріч із Маєю Санду	езпечити експорт електроенергії до Молдови під час опалювального сезону з урахуванням ная
17	льщі провели телефонну розмову з Президентом США Дональдом Трампом	ності закінчувати війну. Володимир Зеленський закликав збільшити тиск на Росію та запровади
18	ни, якби Путін не злякався приїхати до Туреччини – Володимир Зеленський	лідерів. Володимир Зеленський зазначив, що впродовж дня інформуватиме партнерів про те, ц
19	Володимир Зеленський провів зустріч із Реджепом Таїпом Ердоганом	ленський і Реджеп Таїп Ердоган обговорили розвиток двосторонньої співпраці між державами,
20	Президент України провів телефонну розмову з Президентом Туреччини	цього. Володимир Зеленський і Реджеп Таїп Ердоган домовилися про подальшу спільну робо
21	а України: Президент відзначив воїнів найвищими державними нагородами	рім того, Глава держави відзначив орденами «За мужність» III ступеня та княгині Ольги III ступ
22	Володимир Зеленський провів розмову з Папою Римським Левом XIV	Зеленський і Папа Римський Лев XIV домовилися підтримувати контакт і спланувати особисту
23	меччини, Польщі, Великої Британії та України за підсумками зустрічі в Києві	ронної стійкості та інвестиції у виробництво озброєнь як в Україні, так і в європейських країнах
24	до ЄС: Володимир Зеленський провів переговори з Емманюелем Макроном	римку Франції в ухваленні необхідних політичних рішень на рівні Євросоюзу та відкриття перш
25	мир Зеленський і Кір Стармер підсумували сьогоднішні зустрічі з партнерами	новим елементом гарантій. Серед інших важливих тем зустрічі – подальше посилення оборонн
26	Британії провели спільну прогулянку Києвом після засідання коаліції охочих	фійського собору. Весь час лідерів супроводжувала екскурсовод, яка розповіла про минуле й с
27	повне й безумовне припинення вогню щонайменше на 30 днів – Президент	часу відчуття, що весь вільний світ об'єднаний. І це не лозунги, а рішення, які ми ухвалили»,
28	подимир Зеленський: Наше спільне завдання – змусити Москву зупинитися	кими та чіткими. Безумовне припинення вогню означає безумовне. І нічого більше», – підсумув
29	ння зустріч лідерів України, Франції, Великої Британії, Німеччини та Польщі	тиску на Росію, гарантії безпеки та важливість координації всіх зусиль і спільних кроків зі Спол
30	раїни, перша леді та європейські лідери вшанували пам'ять загиблих воїнів	ть загиблих воїнів хвилиною мовчання. Європейські лідери прибули сьогодні до Києва для учас
31	пинення вогню, бо тільки так може розпочатися справжній мир – Президент	ндів Рубен Брекельманс. Президент подякував за участь України у JEF та можливість поєдна
32	а буде притягнута до відповідальності за цю війну – Володимир Зеленський	ри правосуддя та відзначив зусилля Ради Європи й Нідерландів, а саме Гааги за готовність ст
33	раїни провів телефонну розмову з Президентом США Дональдом Трампом	магати. Він також наголосив на тому, що підтримує припинення вогню. Лідери України та США
34	Володимир Зеленський провів розмову з Фрідріхом Мерцом	ів на найближчий час. Глава держави подякував Німеччині за підтримку в захисті життів украї

Figure 10: Structure of the output CSV file

Software for thematic modelling of Ukrainian-language texts based on the Latent Dirichlet Allocation (LDA) algorithm has been developed. The developed system allows you to automatically identify meaningful topics in a collection of texts, interpret them through keywords, and also classify new documents according to the built model. A feature of the implementation is the use of the Stanza library for the lemmatisation of Ukrainian texts, which provides deep linguistic data processing. The program covers the entire cycle of thematic modelling: from collecting and pre-processing texts to building an LDA model, assessing its quality using coherence metrics, and visualising the resulting topics. In the process of implementation, mechanisms for automatic generation of conventional names of topics were also implemented, which significantly facilitates the perception of the results of the analysis. The functionality of the program includes the ability to save processed texts, reuse models, view an interactive topic map, and classify new texts in real time. It has been proven that the model is capable of detecting thematic clusters with high coherence, which indicates its efficiency and accuracy. Thus, the goals of the work have been achieved. The developed software is a universal tool for text data analytics. It can be used to solve practical problems in the fields of journalism, public administration, education, and social sentiment research.

6. Results

Now, in the modern information space, it is essential to be able to quickly analyse large amounts of text data and isolate the main topics and content areas from it. One of the key approaches to solving this problem is thematic modelling of texts, which allows you to automatically detect the hidden structure of information in a large corpus of documents without manual mark-up. This approach is based on the use of machine learning algorithms, in particular, Latent Dirichlet Allocation (LDA), which allows you to break down texts into topics based on statistical patterns in word distribution. Within the framework of the previous work, full-fledged software for thematic modelling of Ukrainian-language texts was implemented. The implementation included the stages of pre-processing of texts, lemmatization using the stanza library, construction of a dictionary of terms, creation of a corpus in the Bag-of-Words format, training of the LDA model, output of topic keywords, evaluation of the quality of the model by the coherence metric (cv), generation of conditional names of topics and classification of new texts. The study is devoted to checking the operability of the developed software tool by running **a control case**. Such an example allows you to make sure that all modules of the system function in a coordinated manner, the results of the simulation correspond to the content of the text, and the system correctly classifies new documents according to the topics that were discovered during the training. The analysis of the control example allows you to confirm that the results obtained are logical, meaningfully relevant, and correspond to the task. Thus, the purpose of this work is to launch and analyse a control case demonstrating the full cycle of the software - from loading a new text to determining its topic, with the output of topic keywords, probabilistic distribution and interpretation of results.

The purpose of the control example is to check the operability of the software for thematic modelling of texts. For this purpose, a test task is formed, which should reflect the key functionality of the system for determining the subject matter of the Ukrainian-language text on the basis of the already trained LDA model. The user enters a new Ukrainian-language text that was not included in the educational building, and the system should:

1. Carry out full pre-processing of the text (cleaning, tokenisation, lemmatisation).
2. Convert text to a numeric format according to the already built dictionary.
3. Transmit text to the trained LDA model.
4. Obtain a probabilistic distribution of topics identified in the previous analysis.
5. Identify the topic with the highest probability.
6. Output keywords and the automatically generated name of this topic.

To train the thematic model and further test the software, a corpus of Ukrainian texts, collected from open sources, in particular, from the official website of the President of Ukraine, was used. The data is from news, public speeches, event reports, international meetings, decrees, and other documents covering socio-political topics. At the initial implementation stage, the first dataset of approximately 300 documents was created. This set made it possible to check the correctness of the main modules of the system – word processing, dictionary construction, creation of a corpus in the Bag-of-Words format, model training, knowledge base construction and primary classification. However, in the analysis process, it was found that the model trained on this set showed insufficient topic resolution, and the coherence (topic quality metric) was below the desired level. The topics were often mixed, vaguely defined, or too general. In order to improve the quality of the model and expand the thematic coverage, a new, significantly larger corpus was created. The second dataset, which was formed as a result, consisted of more than 830 documents, which made it possible to provide better statistical representativeness of words and contexts. The new set of texts covered a wide range of topics: international politics, internal governance, educational issues, commemoration of historical memory, humanitarian initiatives, etc. The extended corpus was used for the final training of the LDA model, the classification of texts and the execution of a control example within this work.

	title	
320	В Офісі Президента відбулася зустріч із делегацією Бюро національної безпеки Польщі	ий та Дар'юш Луковський приділили питанням співпраці двох країн на шляху України до членства в
321	бургу в тематичних міжнародних заходах для реалізації пунктів Формули миру – Ігор Жовква	у підтримку України та надану практичну допомогу з початку повномасштабного російського вторг
322	У Києві відкрили виставку, присвячену участі українських спортсменів в Олімпійських іграх	комітет та асоціація «Дивись українське!». Мета заходу – продемонструвати волю українців до пер
323	иторії тимчасово окупованого півострова Крим – рішення Європейського суду з прав людини	рівника Офісу Президента Ірина Мудра та відзначила роботу команди уповноваженого України в
324	підготовка до саміту НАТО: Андрій Єрмак провів телефонну розмову з Джейком Салліваном	с розмови окрему увагу приділили підготовці до саміту НАТО, який відбудеться 9–11 липня у Ваши
325	ьща провели фінальний раунд переговорів щодо укладення двосторонньої безпекової угоди	раїни, яку ухвалили торік 12 липня. Загалом наша держава уклала вже 17 двосторонніх безпекови
326	Андрій Єрмак обговорив із радником Прем'єр-міністра Канади реалізацію Формули миру	адою, а також організацію низки тематичних конференцій, присвячених реалізації пунктів Формули
327	їна та Ірландія обговорили підсумки Саміту миру та майбутню двосторонню безпекову угоду	зав ірландській стороні за незмінну підтримку України, зокрема на шляху до членства нашої країни
328	и Глобального саміту миру з радником із питань національної безпеки Прем'єр-міністра Індії	'єр-міністра Індії відвідати Україну з візитом. Аджит Кумар Довал зазначив, що опрацює таку можл
329	Україна та ЄС завершили роботу над текстом безпекових гарантій	завершили узгодження тексту безпекової угоди та домовилися про її підписання вже найближчим
330	оворив із радником Президента ПАР підсумки Саміту миру та розвиток подальших контактів	ли подальший план контактів на найвищому рівні між Україною та Південно-Африканською Республ
331	В Офісі Президента обговорили реформування правничої освіти	и. Реформа має на меті покращити якість юридичної освіти та узгодити її з європейськими станда
332	ювернулася на перші шпальти всіх видань світу – Андрій Єрмак про результати Саміту миру	вимірах – і в матеріальному, і в нематеріальному, і в кримінальному зокрема», – зазначив Андрій Є
333	ни РФ в Україні – на Саміті миру відбулася окрема дискусія щодо гуманітарного виміру війни	йському полоні, зазнають систематичних тортур. Є підтвердження щонайменше 14 тисяч таких ви
334	одні Глобального саміту миру відбулася серія заходів у межах ініціативи Bring Kids Back UA	аміту миру у Швейцарії 15–16 червня. Звільнення всіх полонених і депортованих – серед провідни
335	ина Мудра під час слухань ЄСПЛ щодо порушень РФ прав людини на окупованих територіях	ордонних судових органів установив, що території на сході України були окуповані з 11 травня 201
336	Андрій Єрмак за дорученням Президента України відвідав Італію та Ватикан	фраструктури до зимового періоду, а також вивчення можливостей спільного виробництва в сектор
337	Україна та ЄС провели черговий раунд переговорів щодо безпекової угоди	овно обговорили текст проекту угоди та затвердили алгоритм подальших дій на найближчу перспе
338	В Офісі Президента відбулася міжвідомча нарада щодо ратифікації Римського статуту	оли національні суди держави не можуть або не хочуть забезпечувати відповідальність за певні зл
339	овідних світових університетів та освітніх організацій просування Української формули миру	єкти інфраструктури, зокрема зруйнувала 340 закладів освіти та суттєво пошкодила понад 3400 об
340	Офісі Президента відбулася зустріч із представниками студентських і молодіжних організацій	рганізували Експертна рада студентства та Молодіжна рада при Міністерстві закордонних справ Ук
341	ндрій Єрмак обговорив із президентом Всесвітньої боксерської ради підтримку Саміту миру	ів України, тому що всі ви є героями. І наші серця – разом з Україною», – акцентував Маурісіо Суле
342	В Офісі Президента відбулася зустріч із заступником міністра фінансів США	ні підстави. Сполучені Штати мають на меті розпочати роботу над якомога швидшим їх розблокува
343	Україна та Чилі поглиблюють співпрацю для повернення викрадених РФ українських дітей	країнських дітей. Він також висловив сподівання, що країна відіграватиме активну роль у роботі ко
344	алом Ноєм Харарі важливість привернення якнайбільшої уваги до Глобального саміту миру	ння справедливого миру і зробить усе можливе, щоб допомогти Глобальному саміту миру досягти
345	Аргентина планує долучитися до Міжнародної коаліції за повернення українських дітей	с всі аспекти розв'язання питань щодо викрадення росіянами українських дітей та повернення їх д
346	йснила всі чотири законодавчі кроки для початку переговорів про вступ до ЄС – Ігор Жовква	и також обмінялися думками щодо підготовки до інавгураційного Саміту миру 15–16 червня у Шве
347	ен будь-який мир, нам потрібен справедливий мир – Андрій Єрмак під час саміту UNITED24	який працює і на Глобальному Півдні, і на Глобальній Півночі, де завгодно», – підсумував Сергій П
348	Україна та Греція найближчим часом підпишуть безпекову угоду	боку Греції початку переговорів про вступ нашої країни до ЄС, а також наближення до членства в
349	лідерів молодіжного руху долучитися до поширення інформації про Глобальний саміт миру	аді показати: сьогодні сильні лідери спроможні ухвалювати сильні рішення», – резюмував Андрій Є
350	Андрій Єрмак провів телефонну розмову з Джейком Салліваном	гьбі з повномасштабною російською агресією, а також за оборонний пакет, який уже працює на пол
351	АТО зараз було б життєво важливим кроком для зміцнення безпеки Європи – Андрій Єрмак	обочої групи з безпекових питань та євроатлантичної інтеграції України, опублікованому минулого
352	Україна та Швеція розпочали переговори щодо укладення двосторонньої безпекової угоди	годили подальші кроки з метою забезпечення якнайшвидшої його фіналізації та подальшого підпи
353	ошуривати ідею Саміту миру для досягнення справедливого миру в Україні – Андрій Єрмак	и ім. П. І. Чайковського Сунь Цин зазначив, що важливо реалізовувати пункти Української формули

Figure 11: First dataset

	title	
797	ся з Президентом Фінляндії та прем'єр-міністрами Данії, Норвегії та Ісландії	Норвегії та Ісландії за підтвердження участі в Саміті та готовність активно працювати над імплементац
798	Спільна заява третього саміту Україна – Північна Європа в м. Стокгольмі	зпекові угоди. На шляху до їх реалізації нас надихають наполегливість і дух українського народу, солда
799	Україна уклала безпекову угоду з Норвегією	ю, Данією, Канадою, Італією, Нідерландами, Фінляндією, Латвією, Іспанією, Бельгією, Португалією, Шве
800	безпеки та довгострокову підтримку між Україною та Королівством Норвегія	реважну силу. За Україну: Володимир Зеленський, Президент За Королівство Норвегія: Йонас Гар Стер
801	Україна й Ісландія уклали безпекову угоду	ю, Данією, Канадою, Італією, Нідерландами, Фінляндією, Латвією, Іспанією, Бельгією, Португалією, Шве
802	цтво у сфері безпеки та довгострокову підтримку між Україною та Ісландією	матиме переважну силу. За Україну: Володимир Зеленський, Президент За Ісландію: Б'ярні Бенедіктсс
803	Україна та Швеція підписали безпекову угоду	ю, Францією, Данією, Канадою, Італією, Нідерландами, Фінляндією, Латвією, Іспанією, Бельгією, Порту
804	Угода про співробітництво у сфері безпеки між Україною та Швецією	ст матиме переважну силу. За Україну: Володимир Зеленський, Президент За Швецію: Ульф Крістерсс
805	Володимир Зеленський зустрівся з Президентом Португалії	оюза приділили інтеграції України до ЄС і НАТО. Президент України подякував за підтримку нашої краї
806	ніціативах допомагає Україні встояти проти російських ударів – Президент	у також нагадав про участь Португалії в артилерійській коаліції. Крім того, країна планує брати участь у
807	Україна уклала двосторонню безпекову угоду з Португалією	о Британією, Німеччиною, Францією, Данією, Канадою, Італією, Нідерландами, Фінляндією, Латвією, Іс
808	Угода про співробітництво у сфері безпеки між Україною та Португалією	матиме переважну силу. За Україну: Володимир Зеленський, Президент За Португалію:Луїш Монтенег
809	ознайомився з процесом підготовки до використання літаків F-16 в Україні	ажаю вам гарного закінчення навчання. Дуже дякуємо вам, розраховуємо на вас та чекаємо вдома», –
810	Відбулася аудієнція Президента України у Короля бельгійців Філіпа	еленський запросив Його Величність Короля та Її Величність Королеву бельгійців відвідати Україну у зр
811	ьогоріч: результати зустрічі Президента України та Прем'єр-міністра Бельгії	цієї війни є вирішальним для наших європейських інтересів, цінностей та безпеки», – підсумував Прем
812	Україна та Бельгія уклали угоду про гарантії безпеки	з Великою Британією, Німеччиною, Францією, Данією, Канадою, Італією, Нідерландами, Фінляндією, Л
813	безпеки та довгострокову підтримку між Україною та Королівством Бельгія	матиме переважну силу. Від України: Володимир Зеленський, Президент Від Бельгії: Александр Де Кр
814	Президент зустрівся з головами палат і фракцій парламенту Іспанії	ентів, зокрема держав Латинської Америки. Володимир Зеленський запросив іспанських парламентарі
815	Президент України мав аудієнцію в Короля Іспанії Феліпе VI	нт України запросив Його Величність Феліпе VI та Її Величність Королеву Летицію відвідати Україну у зр
816	озвитку технологій додадуть сили Україні й Іспанії – Володимир Зеленський	к він зазначив, що Іспанія вже підготувала нову партію танків Leopard та снарядів, яких потребують Збр
817	Україна уклала двосторонню безпекову угоду з Іспанією	нні угоди з Великою Британією, Німеччиною, Францією, Данією, Канадою, Італією, Нідерландами, Фінл
818	Угода про співробітництво у сфері безпеки між Україною та Іспанією	України: Володимир Зеленський, Президент України За Іспанію: Педро Санчес Перес-Кастехон, През
819	т привітав із професійним святом працівників і працівниць Держспецзв'язку	і Ольги III ступеня, медалями «За військову службу Україні», «Захиснику Вітчизни» та «За бездоганну с
820	Президент зустрівся з маршалком Сенату Республіки Польща	юзом, зокрема щодо важливості майбутнього польського головування для реалізації пріоритетів Україн
821	Президент оглянув зруйновану російським ударом типографію в Харкові	ку друкували близько третини від усієї кількості книжок в Україні. Зокрема, приблизно 40% становили
822	о опалювального сезону на Харківщині: Президент провів нараду в Харкові	о нового в Харкові чи області. Володимир Зеленський доручив Прем'єр-міністру опрацювати реалізаці
823	ція та Північна Македонія представлені на Глобальному саміті миру	пи текст двосторонньої безпекової угоди, яку ми підпишемо за першої можливості», – підсумував Волод
824	орив з амбасадорами UNITED24 посилення міжнародної підтримки України	депортованих дітей. І, звичайно, ваш голос і ваша підтримка будуть надзвичайно важливими», – наголо
825	осійського терору – Президент із нагоди другої річниці заснування ініціативи	ахист загальнолюдських цінностей, і на абсолютно практичну, дієву допомогу державі в час війни», – за
826	ащенням, і за повагою – Президент привітав воїнів із Днем морської піхоти	ідзнаки «За мужність та відвагу» цьому ж батальйону та 1-му окремому батальйону морської піхоти 36
827	Австрія, Албанія та Чилі підтвердили участь у Глобальному саміті миру	єднатися до міжнародної місії з розмінування територій та повернення незаконно депортованих Росією
828	ття наближають перемогу – Президент під час зустрічі з молодими вченими	ідбувся цьогоріч у березні в Тунісі. На ньому представники Малої академії наук здобули 21 нагороду у
829	Президент зустрівся з главою МЗС Німеччини	тними проєктами. Це вже сьомий візит Анналени Бербок до України з початку повномасштабного росій
830	часна техніка та вчасна допомога партнерів – Президент в інтерв'ю Reuters	еретинати кордон. А щоб протистояти Росії в небі, Україні, за словами Глави держави, потрібно щонай

Figure 12: Second dataset

Each text in the dataset is saved in .csv format, in the text column. Additional metadata, such as headers or dates, is not used in the model. Thus, all texts underwent the same processing cycle, which ensured the purity of the experiment and the ability to compare the results. The use of two different corpora in the development process made it possible to assess the impact of sample size on the quality of thematic modelling. It also highlights the flexibility and scalability of the software created, which can work efficiently with enclosures of different sizes.

To check the functionality of the developed software, a separate fragment of Ukrainian text was selected, which was not included in the educational building. This approach allows you to objectively assess the ability of the model to generalise - that is, the ability to apply the formed topics to new, previously unknown texts. A control example simulates a real situation when the user submits an arbitrary text for input and expects the system to correctly recognise its content. The selected fragment refers to the commemoration of the victims of political repression and is a typical example of official political communication. It has a clear thematic focus and contains specific vocabulary that allows you to test the model's ability to identify keywords and classify the document towards the relevant topic. The text was taken from an open source and was not included in the training dataset in advance, which guarantees the fairness of the test.

```
After the inaugural mass, Pope Leo XIV held an audience with President of Ukraine
Volodymyr Zelenskyy and First Lady Olena Zelenska, who became the first for heads of
state. The President congratulated Pope Leo XIV on the beginning of his pontificate and
noted that he is a hope for millions of people who want peace.
"""
```

```
new_text = """
Після завершення інавгураційної меси Папа Римський Лев XIV провів аудієнцію із Президентом України Володи
"""

result = classify_new_text(new_text, lda_model, dictionary)

# Виводимо найімовірніші теми
for topic_num, prob in result:
    print(f"Тема {topic_num + 1} – ймовірність: {prob:.2%}")
```

Figure 13: Text for the test

This example was specifically selected as a control example, since its theme potentially correlates with one or more topics formed by the LDA model (in particular, with issues related to historical memory, political repression or state policy in the field of culture). The following sections will provide a step-by-step analysis of the processing of this fragment, the results of classification, and an assessment of how the model has determined its topic correctly. A control fragment of the text was submitted for input to the software to check the full cycle of its processing and classification. After loading the text, the system automatically carried out all the stages of analysis in accordance with the logic embedded in the architecture of the software tool. In the first step, text pre-processing is performed, which includes lowercase, tokenisation, filtering of service words, and lemmatisation using the Stanza library. It allows you to bring the text to a unified form, where each word is represented in its basic grammatical form. For example, the phrase "honoured the memory of the dead" after lemmatisation turns into a sequence of lemmas "honour", "memory", "deceased". Next, the cleaned text is transformed into a numeric format using a pre-saved dictionary. To do this, each lemma is replaced with a corresponding numerical identifier, and the frequency of its appearance in the text is recorded in the Bag-of-Words format. This format allows you to present

text as a vector that the model can interpret as input for thematic analysis. The third step is to transfer the processed text to the trained LDA model, which conducts the classification. The model returns the probability distribution of topics formed in the process of previous training. As a result, a list of topics with corresponding probability values is obtained. The topic is most likely to be interpreted as the main one to which the input text belongs. At the final stage, the system displays the topic ID, a list of its keywords, and the generated conditional name formed on the basis of the detected topic semantics. It allows the user not only to see the numerical results of the classification, but also to interpret them understandably. Thus, the submitted text goes through a complete cycle of processing: from natural language to a formalised topic with interpretation. It confirms the ability of the software to correctly identify the subject of a new document based on an already trained model.

In the process of developing software for thematic modelling of texts, there was a need to optimise the performance of the pre-processing subsystem. One of the key elements at this stage is the filtering of stop words - that is, those tokens that do not carry a semantic load, but significantly increase the amount of processing during lemmatisation and vocabulary construction. Initially, we used a complete list of Ukrainian stop words, containing more than 300 elements. However, when tested on a full case with more than 800 documents, the processing time exceeded 150 minutes, which is completely inefficient for practical use. In view of this, its own optimised list has been created, which includes only the most used service words - about 50. It made it possible to reduce the processing time to 51 minutes without significantly losing the quality of the topics.

Figure 14: Graph of the influence of the size of the list of stop words on the time of processing texts, where green is a smaller list of stop words, and red is the complete list of stop words

This graph shows how the size of the stop word list affects the processing time of texts during thematic modelling. As you can see, when using a smaller custom list, the processing of the entire case took 51 minutes, while when using the complete list of stop words, it took more than 150 minutes. It is because a higher number of stop words significantly increases the filtering and processing time of each token in the text, especially when processing involves lemmatisation. In this regard, in order to maintain the effectiveness of software execution, it was decided to use a limited but relevant list of the most frequent service words. It made it possible to significantly reduce the processing time without a critical loss of simulation quality. This analysis confirms that thoughtful optimisation during the pre-processing phase has a significant impact on the overall performance and efficiency of the system.

For the convenience of the user and control over the execution of the software, the output of the progress of word processing in real time was implemented. It became essential after expanding the corpus of texts to more than 800 documents, as the pre-processing time (lemmatisation, filtering, and tokenisation) increased to tens of minutes. To avoid a situation where the user does not understand whether the program is "frozen" or really working, a progress bar was added using the tqdm library, which displays a dynamic scale with the number of documents already processed. It allows you to visually observe the progress of processing, estimate the pace of execution and navigate the remaining time until completion. In this way, the output of execution progress has

increased the clarity, predictability, and usability of the system, which is an integral part of front-end interaction even in console applications.

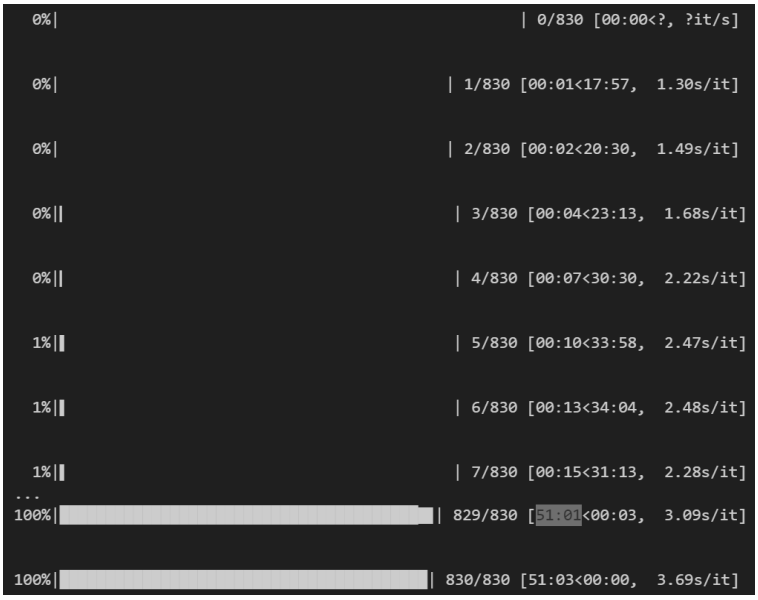

Figure 15: Output of the Word Processing Process

To optimise performance and avoid re-wasting time on text processing, a mechanism for saving already processed data has been implemented. It allows the software to run much faster when reused, especially in the context of experiments, testing models, or changing classification parameters. After the pre-processing step is completed, all cleaned and lemmatised texts are automatically saved as a serialised object in .pkl (pickle) format. In particular, the processed_texts.pkl file stores a list of tokenised texts that have already passed all stages of pre-processing: lowering, removing stop words, lemmatisation, etc. In the future, when the system starts, the program first checks whether the file with the processed data exists. If the file is found, the data is loaded from the disk, and there is no need to process more than 800 documents again, which can take up to an hour. This approach provides significant resource savings and improves user experience, especially in environments with limited execution time, such as during demonstrations, training, or research.

```python
#Збереження результату у файл
with open("processed_texts.pkl", "wb") as f:
    pickle.dump(processed_texts, f)


#для загрузки результату
with open("processed_texts.pkl", "rb") as f:
    processed_texts = pickle.load(f)
```

Figure 16: Save processed text.

In the process of developing a system of thematic modelling of texts, it was decided to use machine learning not only to build the model itself, but also to optimally select the number of topics. It is critically important because too few topics can lead to generalisation and loss of content, and too large a number of issues can lead to excessive division of texts, which reduces the quality of classification.

```python
def compute_coherence_values(dictionary, corpus, texts, start, limit, step):
    coherence_scores = []
    models = []
    for num_topics in range(start, limit + 1, step):
        model = LdaModel(
            corpus=corpus,
            id2word=dictionary,
            num_topics=num_topics,
            random_state=42,
            passes=10,
            alpha='auto'
        )
        coherence_model = CoherenceModel(
            model=model,
            texts=texts,
            dictionary=dictionary,
            coherence='c_v'
        )
        score = coherence_model.get_coherence()
        coherence_scores.append(score)
        models.append(model)
        print(f"num_topics={num_topics} → когерентність={score:.4f}")
    return models, coherence_scores

winsound.Beep(1000, 1000)
```

```python
# Налаштування діапазону
start = 12
limit = 40
step = 1
```

Figure 17: Selection of the best hyperparameters

For each of the models, a coherence metric (in particular, c_v) was calculated, which shows how logically the topics are related in terms of the semantic proximity of words num_topics. The model analysed the quality of the issues constructed, and among all the options, the number of topics that provided the highest coherence was chosen. Thus, the decision was not made manually, but on the basis of an objective indicator of the quality of the model, calculated during the training. Thanks to this approach, it was possible to achieve a more stable, interpreted, and high-quality thematic model that confirms the effectiveness of the use of machine learning methods in the tasks of thematic analysis of texts. In order to determine the optimal number of topics for building a thematic model, a series of experiments was conducted using machine learning and coherence metrics (in particular, c_v). In the course of these experiments, 29 LDA models were built with a different number of topics from 12 to 40. Based on each of the models, the coherence of the indicator was calculated, reflecting how logically the words in the topic are related to each other from the point of view of real language usage. The visualisation of the results was presented in the form of a line graph, where the number of topics is displayed along the X axis, and the coherence values are displayed along the Y axis. The graph clearly shows the fluctuations in the quality of the models. The highest coherence values were achieved for the following configurations

- num_topics=12 ⟶ coherence = 0.5135
- num_topics=22 ⟶ coherence = 0.5167
- num_topics=27 ⟶ coherence = 0.5102
- num_topics=28 ⟶ coherence = 0.5025

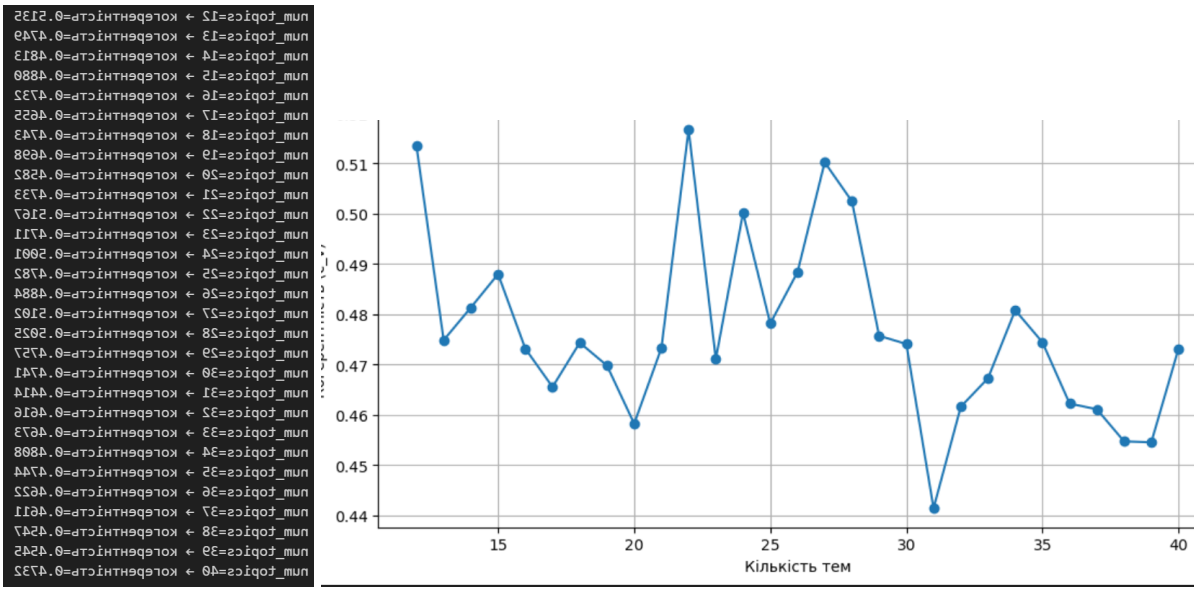

Figure 18: Analysis of hyperparameters – selection of the optimal number of topics

It indicates that these configurations describe the topics in the corpus in the most balanced way, providing high semantic coherence of topic keywords. As can be seen from the graph, too many issues lead to a decrease in coherence, since the model "blurs" the context between too many problems. Based on this analysis, the optimal number of topics was selected - 22, which provides the highest coherence within the framework of the experiment. Thus, the process of choosing the number of issues was not implemented manually, but based on a quality metric that follows the principles of reasonable customisation of machine learning models.

```
# 🔹 Навчання LDA-моделі
lda_model = LdaModel(
    corpus=corpus,
    id2word=dictionary,
    num_topics=22,          #кількість тем
    random_state=42,
    passes=10,
    alpha='auto'
)

winsound.Beep(1000, 1000)
```

Figure 19: LDA model

This code fragment is implemented at the training stage of the LDA model, which is the basis for the thematic modelling of texts. Its goal is to create a machine model that will be able to detect hidden topics in Ukrainian-language texts based on the joint appearance of words in documents. The parameter num_topics=22 is because I previously conducted an automated coherence analysis and determined that 22 topics provided the highest quality of issues (coherence ≈ 0.5167). Thanks to the passes parameter=10 model, which passes through the entire body 10 times, you can achieve greater stability in the topics. Setting alpha='auto' allows the model to independently adapt the distribution of topics in documents, which is especially useful when working with imperfectly balanced data. Beep(1000, 1000), which is triggered after the completion of the training. It is done for the convenience of observation, since training can take tens of minutes, and the beep helps not to constantly monitor the laptop. This stage is key, because it is here that the model is formed, which will later be:

- classify new texts by topic;

- allow you to visualise the connections between words;
- serve as the basis for the interpretation and generation of topic names.

After completing the training of the Thematic Modelling Model (LDA) based on Ukrainian-language news texts, the system formed 22 topics. Each of the issues is represented by a set of the most relevant words that have appropriate weights reflecting their significance for a particular topic. These keywords are the result of a probabilistic distribution of words in the corpus and allow you to gain a deeper understanding of the content of each topic. One example is the theme dominated by the words "Ukraine", "President", "Volodymyr", "Zelensky", "support" - indicates political content, in particular related to leadership and international activities. Another topic may include words like "generation", "youth", "culture", which indicate a completely different semantic emphasis. The results obtained make it possible to automatically interpret topics, analyse information flows and structure large volumes of texts. Thematic word distributions are further used to generate topic names, which makes the models more understandable for the user. It also opens up the possibility of classifying new texts: the system can determine which topic the newly received text belongs to, with the corresponding probability. Thus, this stage is critically important in the entire chain of operation of the software tool, because it is on its basis that a knowledge base is formed, which provides all the further functionality of analysis, interpretation and visualisation of text data.

```
Тема 21: 0.016*"римський" + 0.014*"xiv" + 0.013*"папа" + 0.010*"святий" + 0.010*"лев" + 0.009*"мес" + 0.007*"петро"
Тема 13: 0.056*"індія" + 0.014*"індійський" + 0.014*"міністр" + 0.013*"між" + 0.013*"україна" + 0.013*"модя" + 0.010*"культура"
Тема 3: 0.012*"комісар" + 0.010*"оон" + 0.009*"гранді" + 0.009*"філіппо" + 0.009*"верховний" + 0.006*"біженець" + 0.005*"українець"
Тема 18: 0.015*"фінляндія" + 0.010*"обсє" + 0.006*"словацький" + 0.005*"александр" + 0.005*"для" + 0.005*"михал" + 0.005*"валтонен"
Тема 16: 0.030*"рятувальний" + 0.027*"начальник" + 0.027*"державний" + 0.022*"пожежній" + 0.021*"частина" + 0.019*"рятувальник" + 0.019*"дснс"
Тема 6: 0.012*"свій" + 0.011*"народ" + 0.011*"кримськотатарський" + 0.009*"геноцид" + 0.008*"україна" + 0.008*"грузія" + 0.008*"володимир"
Тема 15: 0.019*"україна" + 0.018*"компанія" + 0.014*"президент" + 0.011*"володимир" + 0.011*"зеленський" + 0.010*"також" + 0.009*"міністр"
Тема 19: 0.015*"україна" + 0.013*"зеленський" + 0.013*"бути" + 0.011*"володимир" + 0.011*"президент" + 0.010*"український" + 0.010*"олена"
Тема 4: 0.017*"під" + 0.012*"окупант" + 0.012*"знищити" + 0.011*"час" + 0.011*"рік" + 0.009*"україна" + 0.009*"позиція"
Тема 14: 0.015*"україна" + 0.012*"чехія" + 0.012*"для" + 0.009*"бути" + 0.008*"ветеран" + 0.008*"також" + 0.007*"володимир"
Тема 17: 0.025*"україна" + 0.018*"президент" + 0.014*"держава" + 0.013*"зеленський" + 0.013*"володимир" + 0.011*"український" + 0.010*"про"
Тема 10: 0.044*"україна" + 0.013*"підтримка" + 0.012*"угода" + 0.012*"сфера" + 0.011*"сторона" + 0.011*"безпека" + 0.010*"для"
Тема 1: 0.014*"україна" + 0.011*"під" + 0.011*"окупант" + 0.011*"російський" + 0.009*"бригада" + 0.009*"воїн" + 0.009*"президент"
Тема 22: 0.022*"україна" + 0.018*"зеленський" + 0.018*"володимир" + 0.016*"президент" + 0.013*"держава" + 0.012*"наш" + 0.008*"глава"
Тема 20: 0.030*"україна" + 0.025*"президент" + 0.017*"володимир" + 0.016*"зеленський" + 0.015*"безпека" + 0.014*"бути" + 0.013*"для"
Тема 7: 0.030*"україна" + 0.015*"президент" + 0.013*"зеленський" + 0.013*"володимир" + 0.011*"український" + 0.011*"цей" + 0.010*"наш"
Тема 9: 0.022*"україна" + 0.015*"для" + 0.014*"бути" + 0.010*"держава" + 0.010*"президент" + 0.009*"цей" + 0.009*"також"
Тема 12: 0.021*"україна" + 0.017*"бути" + 0.012*"президент" + 0.012*"володимир" + 0.012*"для" + 0.012*"зеленський" + 0.011*"мир"
Тема 2: 0.032*"україна" + 0.022*"президент" + 0.017*"володимир" + 0.017*"зеленський" + 0.015*"держава" + 0.013*"про" + 0.011*"для"
Тема 11: 0.040*"україна" + 0.021*"президент" + 0.020*"зеленський" + 0.020*"володимир" + 0.017*"підтримка" + 0.016*"держава" + 0.012*"також"
```

Figure 20: Learning outcome

The image shows an interactive visualisation of the results of thematic modelling created using the pyLDAvis library. This approach allows you to intuitively understand the structure of the constructed LDA model and assess how clearly the topics are delineated and which words are the most characteristic for each of them. The visualisation consists of two parts: the left pane shows a map of topics, and the right pane shows a list of the most relevant terms for the selected topic. On the left side of the visualisation, the so-called "Intertopic Distance Map" is displayed, which demonstrates how topics are arranged in vector space. Each circle represents a different topic, and its size reflects the proportion of documents related to that topic. The distance between the circles indicates the similarity of the topics: the closer the circles, the more similar the topics in content, and if the circles do not intersect, this shows a clear separation of topics. For example, the largest circle on the graph is topic 1, which occupies the largest share in the corpus of texts. The right pane lists the 30 most important terms for the selected topic. Light blue bars indicate the total frequency of use of a word in all texts, while red bars indicate the frequency of this word in the selected topic. It allows you to see which words are really relevant to a particular topic and not just frequently used in the corpus. In our case, topic one is characterised by the phrase "Ukraine", "president", "Zelensky", "Volodymyr", "support", "state", etc., which indicates political topics related to state power and the country's leadership. This type of visualisation is beneficial for analysing the

quality of the model, interpreting the content of topics, and later use in the user interface or reports. It allows not only an analyst, but also an ordinary user without deep knowledge of machine learning to quickly understand what each topic is about and how well the model divided the topics of the documents.

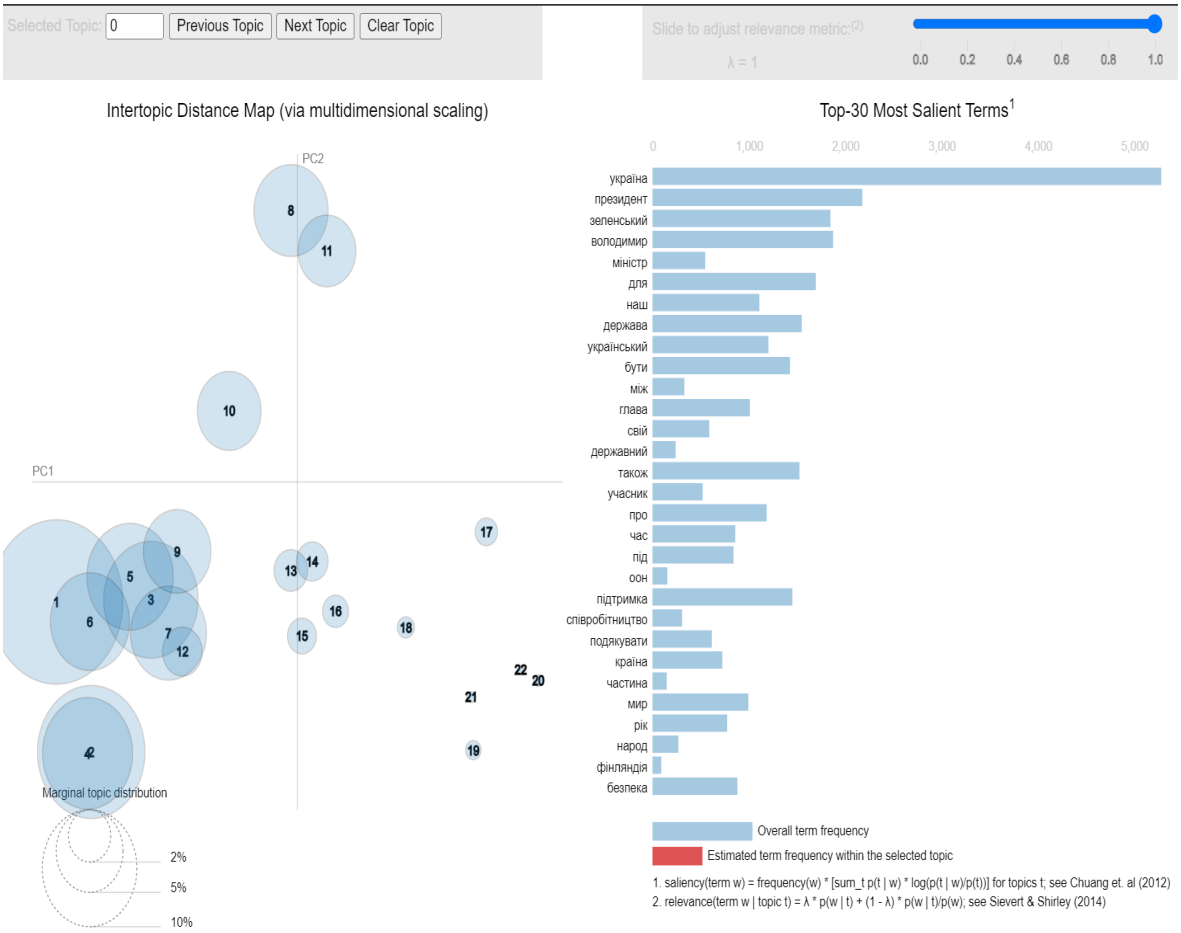

Figure 21: Interactive visualisation of topics

```
Тема 1: 33 документів
Тема 2: 107 документів
Тема 3: 2 документів
Тема 4: 2 документів
Тема 5: 4 документів
Тема 6: 4 документів
Тема 7: 39 документів
Тема 8: 3 документів
Тема 9: 40 документів
Тема 10: 3 документів
Тема 11: 208 документів
Тема 12: 76 документів
Тема 14: 7 документів
Тема 15: 4 документів
Тема 16: 3 документів
Тема 17: 22 документів
Тема 19: 5 документів
Тема 20: 68 документів
Тема 21: 1 документів
Тема 22: 36 документів
```

Figure 22: Distribution of topics

In the course of the implementation of the software for thematic modelling, all documents from the corpus of texts were divided into topics that the trained LDA model defined. It made it possible to see which topics are the most common among the analysed texts, as well as to identify less covered or even highly specialised areas. The image shows the final statistics: each topic corresponds to a certain number of documents. For example, the most significant number of texts, 208, fell into topic 11. It means that this topic is the most representative of the corpus, and its content has the most significant information load. Topics 2 (107 papers), 12 (76 papers) and 20 (68 papers) also have a considerable number, suggesting that the texts are mostly centred around a few leading topics. At the same time, some topics cover only 1-3 documents (for example, topics 21, 4, 3, 10, 16). It may be due to the fact that some texts cover particular events or topics that do not have a broad representation in the general corpus. Such a distribution is proper both for assessing the balance of a data set and for further use in analysis, for example, to identify thematic priorities in news content, to identify topics that require additional attention, or to divide texts into thematic clusters. It also allows you to form an idea of the temperature coverage, which can be used for decision-making in a journalistic, informational or analytical context. After the LDA model formed topics in the form of a set of keywords, there was a need to make them more understandable to a person. After all, a set of words is just a machine representation, from which it is difficult to quickly understand what precisely the topic is about. Therefore, a special module was implemented that automatically generates topic names based on the keywords that characterise them. For example, if among the keywords of the topic there are often "president", "office", "Vladimir", then such a topic can be called "Presidential activity". If the words refer to such concepts as "child", "protection", "rights", the topic is called "Protection of children's rights", etc. Thus, we do not just leave issues in the form of machine combinations of words, but transform them into human-readable titles. It greatly facilitates the perception of modelling results and makes them suitable for practical application both in reports and in interactive text analysis. The user can now easily navigate which topic means which without having to analyse a technical set of words. It is an essential step in "interpreting" the model and bringing the results of machine learning closer to real use.

```
Тема 1: Під і окупант у контексті знищити ➤ під, окупант, знищити, україна, російський, рік
Тема 2: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, держава, про
Тема 3: Час і український у контексті фіцпатрік ➤ час, український, фіцпатрік, інструктор, браян, під
Тема 4: Під і час у контексті бригада ➤ під, час, бригада, україна, російський, група
Тема 5: Україна і норвегія у контексті зеленський ➤ україна, норвегія, зеленський, володимир, мелон, президент
Тема 6: Школа і президент у контексті україна ➤ школа, президент, україна, дитина, штат, володимир
Тема 7: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, український, цей
Тема 8: Україна і ступінь у контексті орден ➤ україна, ступінь, орден, бути, президент, iii
Тема 9: Україна і для у контексті бути ➤ україна, для, бути, президент, держава, зеленський
Тема 10: Україна і сторона у контексті литва ➤ україна, сторона, литва, сполучений, сфера, цей
Тема 11: Україна і президент у контексті зеленський ➤ україна, президент, зеленський, володимир, підтримка, держава
Тема 12: Україна і бути у контексті президент ➤ україна, бути, президент, володимир, зеленський, мир
Тема 13: Нідерланд і каспар у контексті кунінор ➤ нідерланд, каспар, кунінор, японія, мацуда, велдкамп
Тема 14: Людина і удар у контексті україна ➤ людина, удар, україна, начальник, бути, володимир
Тема 15: Бути і виробництво у контексті україна ➤ бути, виробництво, україна, оборонний, компанія, президент
Тема 16: Державний і рятувальний у контексті начальник ➤ державний, рятувальний, начальник, пожежній, частина, область
Тема 17: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, німеччина, держава
Тема 18: Обсє і для у контексті словацький ➤ обсє, для, словацький, юти, фінляндія, валтонен
Тема 19: Україна і бути у контексті рабин ➤ україна, бути, рабин, український, президент, зеленський
Тема 20: Україна і президент у контексті володимир ➤ україна, президент, володимир, зеленський, мир, безпека
Тема 21: Римський і ватикан у контексті святий ➤ римський, ватикан, святий, папа, престол, xiv
Тема 22: Україна і зеленський у контексті володимир ➤ україна, зеленський, володимир, президент, держава, наш
```

Figure 23: Normalise topic names

At the final stage of the work, the function of recognising the topic of third-party text by calculating the probabilities of its belonging to already trained topics was implemented. It made it possible to assess the practical ability of the built LDA model to classify new documents without prior reference to the educational building. An experiment was conducted with a test piece of text that was not part of the training kit. The distribution of topics for this text was analysed separately for two models: the one that was trained on a smaller corpus of about 300 texts and the one based

on an extended corpus of 830 articles. For a larger dataset, the central theme received a weight of 49.62%, which indicates the high confidence of the model in the classification.

On the other hand, on a smaller dataset, the main topic had a similar weight - 48%, but the second topic was almost equal to it - 46%, which may indicate a lower accuracy of the model due to a lack of training examples. The graph below shows a comparison of the distribution of topics between the two models. Visualisation confirms that the growth of the volume of data significantly improves the clarity of classification and reduces the blurring of results. It also reduces the chance of misclassification of text between two nearly equivalent topics. Thus, the increase in the learning corpus directly affects the quality of topic recognition in new documents.

A test run of the software for thematic modelling of Ukrainian-language texts was carried out, which confirmed its operability and compliance with the task. The main goal was to check whether the system built on the basis of the LDA model is able to recognise topics in new texts and provide a meaningful interpretation of the results. In the course of the work, a test task was formulated - to automatically determine the topic of the new Ukrainian-language text. For this, a model previously trained on a large body of news articles was used. Two training options were tested: on a smaller set (approximately 300 documents) and on a much larger set (more than 800 documents). It made it possible to see the impact of the amount of data on the accuracy of the distribution of topics. As the analysis showed, the model trained on a larger dataset demonstrated higher coherence and a more stable probability distribution, which indicates a higher quality of thematic classification. During testing, a complete cycle was implemented: pre-processing of the text with lemmatisation, construction of thematic distribution, interpretation of topics, generation of topic names, and display of results in a convenient form. It was conveniently organised to control the processing execution (through process output and completion signals), save data to avoid re-wasting resources, and visualise the results in the form of diagrams and pyLDAvis graphs. Summing up, it can be argued that the developed software not only demonstrates correct technical implementation but is also able to provide flexible, effective thematic modelling of text data. Such a tool can be helpful for analysts, journalists, researchers, or information systems that require quick orientation in large arrays of Ukrainian-language texts.

7. Discussion

When developing software, coherence (a measure of consistency of topics) is compared when using different amounts of data. For the first dataset (~300 texts), the coherence of the model was 0.462, while after switching to the extended dataset (~830 texts), it increased to 0.516. It suggests that a larger body allows the model to better shape topics - the keywords in them are more related, and the classification results are more resistant to random deviations. Thus, the quality of the model directly depends on the volume of the educational building

Table 25
Performance

Type of stop words	Word count	Preprocessing Time
Hand-picked	~40	~51 min
Advanced (full)	~200+	~150+ min

In order not to repeat the lengthy processing process each time, it is implemented to save the processed case to the processed_texts.pkl file. It allows you to load ready-made data at the subsequent launch of the program, which reduces the waiting time from tens of minutes to several seconds. In addition, visual observation of pre-processing progress via tqdm has been implemented, and a sound signal has been added after the model training is completed, which is convenient for long calculations. To optimise performance and avoid re-wasting time on text processing, a

mechanism for saving already processed data has been implemented. It allows the software to run much faster when reused, especially in the context of experiments, testing models, or changing classification parameters. After the pre-processing step is completed, all cleaned and lemmatised texts are automatically saved as a serialised object in .pkl (pickle) format. In particular, the processed_texts.pkl file stores a list of tokenised texts that have already passed all stages of pre-processing: lowering, removing stop words, lemmatisation, etc. In the future, when the system starts, the program first checks whether the file with the processed data exists. If the file is found, the data is loaded from the disk, and there is no need to process more than 800 documents again, which can take up to an hour. This approach provides significant resource savings and improves user experience, especially in environments with limited execution time, such as during demonstrations, training, or research.

```python
#Збереження результату у файл
with open("processed_texts.pkl", "wb") as f:
    pickle.dump(processed_texts, f)


#для загрузки результату
with open("processed_texts.pkl", "rb") as f:
    processed_texts = pickle.load(f)
```

Figure 24: Save processed text

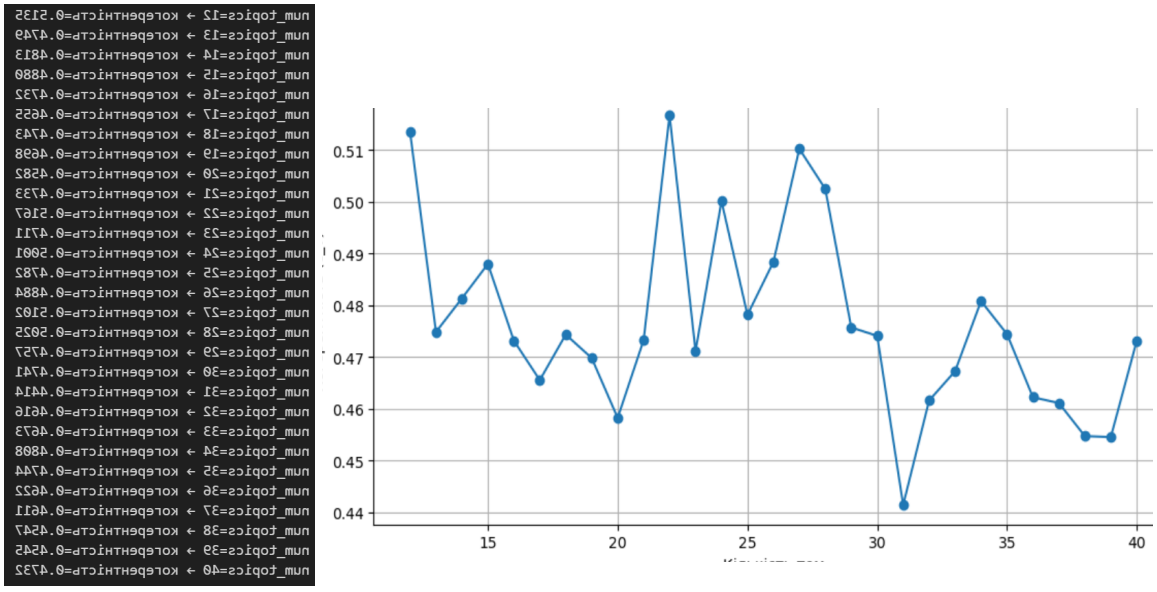

Figure 25: Analysis of hyperparameters - selection of the optimal number of topics

In order to determine the optimal number of topics for building a thematic model, a series of experiments using machine learning and coherence metrics (including c_v) were conducted. In the course of these experiments, 29 LDA models were built with a different number of topics from 12 to 40. Based on each of the models, coherence was calculated - an indicator that reflects how logically the words in the topic are related to each other from the point of view of real language usage. The visualisation of the results was presented in the form of a line graph, where the number of topics is displayed along the X axis, and the coherence value is displayed along the Y axis. The graph clearly shows the fluctuations in the quality of the models.

The distribution of topics for this text was analysed separately for two models: the one that was trained on a smaller corpus of about 300 texts and the one based on an extended corpus of 830 articles. For a larger dataset, the central theme received a weight of 49.62%, which indicates the high confidence of the model in the classification. On the other hand, on a smaller dataset, the main

topic had a similar weight - 48%, but the second topic was almost equal to it - 46%, which may indicate a lower accuracy of the model due to a lack of training examples. The graph below shows a comparison of the distribution of topics between the two models. Visualisation confirms that the growth of the volume of data significantly improves the clarity of classification and reduces the blurring of results. It also reduces the chance of misclassification of text between two nearly equivalent topics. Thus, the increase in the learning corpus directly affects the quality of topic recognition in new documents.

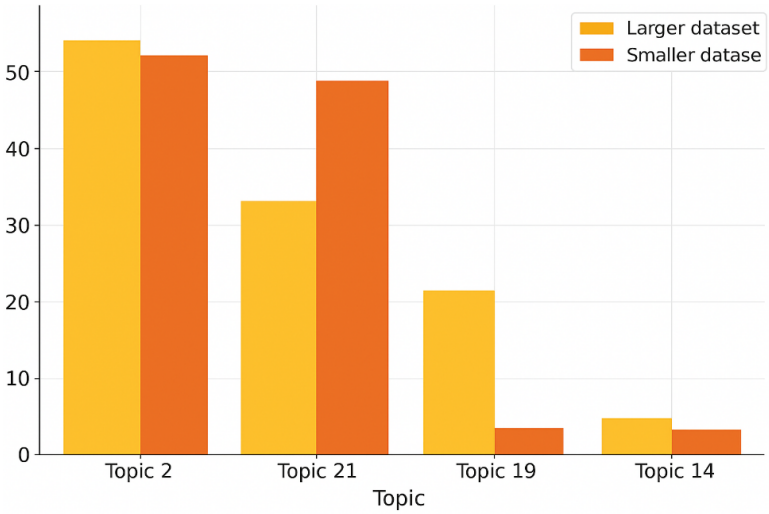

Figure 26: Comparison for different datasets, where yellow – the larger, orange – smaller dataset

```
Тема 2 – ймовірність: 49.62%
Тема 21 – ймовірність: 29.41%
Тема 19 – ймовірність: 15.42%
Тема 14 – ймовірність: 3.90%
```

Figure 27: Topic recognition result

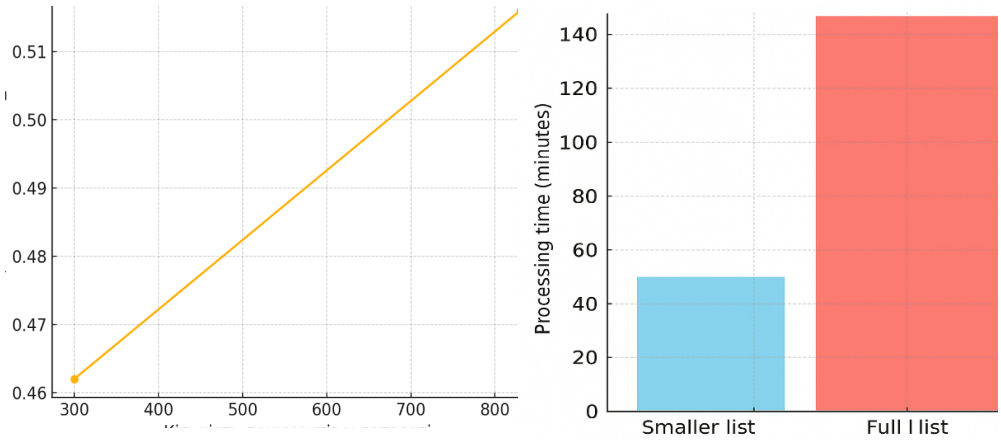

Figure 28: Model dependencies, where on the left is the influence of the dataset volume on coherence, and on the right is the time of text processing depending on the list of stop words

The left graph demonstrates how an increase in the volume of the dataset has a positive effect on the quality of the thematic model. When the first dataset, consisting of ~300 documents, was used, the coherence of the model (a measure of its thematic consistency) was approximately 0.462. After expanding the corpus to more than 800 documents, the coherence value increased to 0.516, indicating an improvement in the quality of the topic classification. The right graph illustrates how the number of stop words affects the processing time of the text. When using a smaller list of stop

words, the process of pre-processing the entire text took approximately 51 minutes. However, an attempt to apply a complete extended list led to a significant increase in duration - more than 150 minutes. It showed me that in order to maintain processing efficiency, you need to find a balance between the depth of text clean-up and performance.

The main goal of this work was not just to check the functionality of the model but also to assess how stable, fast, and qualitatively it works under different conditions. It was found that the quality of thematic modelling directly depends on the volume of the educational building. With an increase in the number of documents from ~300 to more than 800, the coherence of the model increased significantly from 0.462 to 0.516, which indicates better structured and accurate topics. In this way, the model becomes more meaningfully expressive and resistant to mixed themes. Separately, the performance of the system was analysed, particularly the time required for word processing. It turned out that the use of a complete list of stop words dramatically increases the duration of pre-processing from 51 to more than 150 minutes. It was the basis for the decision to use a shortened, optimised list of stop words, which allows you to maintain a balance between processing depth and performance. In addition, a number of technical improvements have been implemented: a progress bar (`tqdm`), a sound signal about completion, and saving processed texts to a file (`pickle`). It made it possible to save time significantly when restarting the program and made interaction with it more comfortable. Thanks to the analysis, it became apparent that the created software is not only functional, but also efficient, scalable and suitable for further use in real tasks of text data analysis. The results of the work confirmed the feasibility of using machine learning for thematic modelling and the importance of correctly adjusting parameters to achieve maximum quality.

8. Conclusions

Software for thematic modelling of Ukrainian-language texts based on the LDA (Latent Dirichlet Allocation) algorithm was designed, implemented and tested. The system was created from scratch, taking into account the peculiarities of the Ukrainian language, the specifics of working with text corpora and the requirements for the interpretation of results for an ordinary user. Several datasets were built: at the first stage, a test case of about 300 documents, and later a full-fledged extended case with a volume of more than 830 documents. It made it possible to conclude the effect of the amount of training data on the quality of the model, in particular, on coherence (which increased from 0.46 to 0.516 with an increase in the corpus). The system covers all the main stages of text analysis: pre-processing (cleaning, tokenisation, lemmatisation via stanza), conversion to numerical format, training a thematic model, building a dictionary of topics, automatic assignment of new texts to topics, as well as generation of conditional names of topics for user convenience. Visualisation of results via pyLDAvis was also implemented, which made it possible to better interpret the topic space and estimate the distances between them. Particular attention was paid to usability: saving processed data (pickle), displaying processing progress via tqdm, sound notifications about the completion of calculations, and optimisation of work with stop words. Thanks to these solutions, the software became not only functional, but also practical in Use. After training the model, the functionality of classifying new (third-party) texts was implemented and tested. The results demonstrate that the system is able to correctly determine the subject matter of even those documents that it has not seen before. The results were compared using two variants of the trained model on smaller and larger datasets. In both cases, the model returned meaningful and logical results, but from a larger case, the results were more stable and more confidently interpreted. This project is essential in terms of the practical application of natural language processing and machine learning methods. It proved that it is possible to effectively perform thematic modelling of Ukrainian-language documents using modern tools (gensim, stanza, pyLDAvis) even without the use of powerful clusters or ample computing resources. It has been confirmed that the quality of the LDA model significantly depends on the hull volume, purity, and quality of pre-processing, optimal selection of the number of topics, and balance between the

completeness of the stop dictionary and the speed of processing. The developed software can be adapted to other languages, extended for more complex corpora, or integrated into larger systems such as web applications, dashboards, or content filtering systems. Prospects for further research:

- Integration of other topic models - such as BERTopic or NMF using modern vector representations (e.g. BERT or FastText). It can increase accuracy and flexibility in defining topics.
- Evaluation of the quality of the model by the user - implementation of feedback mechanisms (for example, if the user agrees or disagrees with the topic assigned to the text).
- Analysis of the dynamics of topics over time - identifying how popular issues in the news stream or publications change over periods.
- Clustering of users or sources - based on the topics they produce or read; it is possible to build recommended systems.
- Deeper coherence research involves various metrics (u_mass, c_npmi) and manual evaluation by experts.

The study made it possible to create a full-fledged, functional and optimised system for thematic analysis of texts in Ukrainian. It combines elements of machine learning, natural language processing, and visual analytics. Work on the project made it possible to deepen practical skills in building NLP models, optimising code, and interpreting results. In addition to the practical result, it was also a meaningful learning experience, forming the basis for more complex research or commercial decisions in the future.

Acknowledgements

The research was carried out with the grant support of the National Research Fund of Ukraine, "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users", project registration number 33/0012 from 3/03/2025 (2023.04/0012). Also, we would like to thank the reviewers for their precise and concise recommendations that improved the presentation of the results obtained.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] D. M. Blei, A. Y. Ng, M. I. Jordan, *Latent Dirichlet Allocation*, *Journal of Machine Learning Research* 3(Jan) (2003) 993–1022. URL: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

[2] R. Egger, J. Yu, *A topic modeling comparison between LDA, NMF, Top2Vec, and BERTopic to demystify Twitter posts*, *Frontiers in Sociology* 7 (2022) 886498. doi:10.3389/fsoc.2022.886498.

[3] M. Grootendorst, *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*, *arXiv preprint* arXiv:2203.05794 (2022). doi:10.48550/arXiv.2203.05794.

[4] F. Bianchi, *Contextualized Topic Models*. URL: https://github.com/MilaNLProc/contextualized-topic-models.

[5] Z. Fang, Y. He, R. Procter, *CWTM: Leveraging contextualized word embeddings from BERT for neural topic modeling*, *arXiv preprint* arXiv:2305.09329 (2023). URL: https://arxiv.org/html/2305.09329v3.

[6] O. B. Petrovych, *Topic modelling of Ukrainian folk songs: A case study on Podillia region*, in: *CS&SE@SW*, (2024) 183–198. URL: https://ceur-ws.org/Vol-3917/paper45.pdf.

[7] C. Maathuis, I. Kerkhof, *The first two months in the war in Ukraine through topic modeling and sentiment analysis*, *Regional Science Policy & Practice* 15(1) (2023) 56–75. doi:10.1111/rsp3.12632.

[8] Published Papers on CTM. URL: https://contextualized-topic-models.readthedocs.io/en/latest/papers.html.

[9] S. Adhya, A. Lahiri, D. K. Sanyal, P. P. Das, *Improving contextualized topic models with negative sampling*, *arXiv preprint* arXiv:2303.14951 (2023). URL: https://aclanthology.org/2022.icon-main.18.pdf.

[10] A. Agrawal, W. Fu, T. Menzies, *What is wrong with topic modeling? And how to fix it using search-based software engineering*, *Information and Software Technology* 98 (2018) 74–88. doi:10.1016/j.infsof.2018.02.005.

[11] D. Sharma, B. Kumar, S. Chand, *A survey on journey of topic modeling techniques from SVD to deep learning*, *International Journal of Modern Education and Computer Science* 9(7) (2017) 50. doi:10.5815/ijmecs.2017.07.06.

[12] B. Ogunleye, T. Maswera, L. Hirsch, J. Gaudoin, T. Brunsdon, *Comparison of topic modelling approaches in the banking context*, *Applied Sciences* 13(2) (2023) 797. doi:10.3390/app13020797.

[13] C. B. Pavithra, J. Savitha, *Topic modeling for evolving textual data using LDA, HDP, NMF, BERTopic and DTM with a focus on research papers*, *Journal of Technology and Informatics (JoTI)* 5(2) (2024) 53–63. doi:10.37802/joti.v5i2.618.

[14] A. Thielmann, A. Reuter, Q. Seifert, E. Bergherr, B. Säfken, *Topics in the haystack: Enhancing topic quality through corpus expansion*, *Computational Linguistics* 50(2) (2024) 619–655. doi:10.1162/coli_a_00506.

[15] A. Mutsaddi, A. Jamkhande, A. Thakre, Y. Haribhakta, *BERTopic for Topic Modeling of Hindi Short Texts: A Comparative Study*, *arXiv preprint* arXiv:2501.03843 (2025). doi:10.48550/arXiv.2501.03843.

[16] V. Vysotska, K. Przystupa, Y. Kulikov, S. Chyrun, Y. Ushenko, Z. Hu, D. Uhryn, *Recognizing Fakes, Propaganda and Disinformation in Ukrainian Content based on NLP and Machine-learning Technology*, *International Journal of Computer Network and Information Security (IJCNIS)* 17(1) (2025) 92–127. doi:10.5815/ijcnis.2025.01.08.

[17] M. Nyzova, V. Vysotska, L. Chyrun, Z. Hu, Y. Ushenko, D. Uhryn, *Smart Tool for Text Content Analysis to Identify Key Propaganda Narratives and Disinformation in News Based on NLP and Machine Learning*, *IJCNIS* 17(4) (2025) 113–175. doi:10.5815/ijcnis.2025.04.08.

[18] R. Lynnyk, V. Vysotska, Z. Hu, D. Uhryn, L. Diachenko, K. Smelyakov, *Information Technology for Modelling Social Trends in Telegram Using E5 Vectors and Hybrid Cluster Analysis*, *International Journal of Information Technology and Computer Science (IJITCS)* 17(4) (2025) 80–119. doi:10.5815/ijitcs.2025.04.07.

[19] V. Vysotska, Z. Hu, N. Mykytyn, O. Nagachevska, K. Hazdiuk, D. Uhryn, *Development and Testing of Voice User Interfaces Based on BERT Models for Speech Recognition in Distance Learning and Smart Home Systems*, *IJCNIS* 17(3) (2025) 109–143. doi:10.5815/ijcnis.2025.03.07.

[20] V. Vysotska, K. Przystupa, L. Chyrun, S. Vladov, Y. Ushenko, D. Uhryn, Z. Hu, *Disinformation, Fakes and Propaganda Identifying Methods in Online Messages Based on NLP and Machine Learning Methods*, *IJCNIS* 16(5) (2024) 57–85. doi:10.5815/ijcnis.2024.05.06.

[21] V. Vysotska, *Linguistic intellectual analysis methods for Ukrainian textual content processing*, in: *CEUR Workshop Proceedings*, vol. 3722 (2024) 490–552. URL: https://ceur-ws.org/Vol-3722/paper25.pdf