

Methodology for Automatic Information Extraction and Summary Generation from Online Sources for Project Funding [†]

Mariya Zhekova 

Department of Computer Systems and Technologies, University of Food Technology, 4000 Plovdiv, Bulgaria; m_jekova@uft-plovdiv.bg

[†] Presented at the 14th International Scientific Conference TechSys 2025—Engineering, Technology and Systems, Plovdiv, Bulgaria, 15–17 May 2025.

Abstract

The summarized content of one or more extensive text documents helps users extract only the most important key information, instead of reviewing and reading hundreds of pages of text. This study uses extractive and abstractive mechanisms to automatically extract and summarize information retrieved from various web documents on the same topic. The research aims to develop a methodology for designing and developing an information system for pre- and post-processing natural language obtained through web content search and web scraping, and for the automatic generation of a summary of the retrieved text. The research outlines two subtasks. As a first step, the system is designed to collect and process up-to-date information based on specific criteria from diverse web resources related to project funding, initiated by various organizations such as startups, sustainable companies, municipalities, government bodies, schools, the NGO sector, and others. As a second step, the collected extensive textual information about current projects and programs, which is typically intended for financial professionals, is to be summarized into a shorter version and transformed into a suitable format for a wide range of non-specialist users. The automated AI software tool, which will be developed using the proposed methodology, will be able to crawl and read project funding information from various web documents, select, process, and prepare a shortened version containing only the most important key information for its clients.



Academic Editors: Nikola G. Shakev,
Valyo N. Nikolov, Nikolay
R. Kakanakov and Sevil
A. Ahmed-Shieva

Published: 11 July 2025

Citation: Zhekova, M. Methodology for Automatic Information Extraction and Summary Generation from Online Sources for Project Funding. *Eng. Proc.* **2025**, *100*, 44.
<https://doi.org/10.3390/engproc2025100044>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: summarization; retrieving information; artificial intelligence; abstract generation

1. Introduction

The automatic generation of summarized content from documents has intrigued researchers over the past five decades. As digital information continues to grow exponentially, people increasingly seek ways to reduce the time and optimize the means for analyzing and understanding this information. Summarizing content means extracting only the most essential and useful information from large text documents, creating a shortened, compressed version of the original material instead of reading through hundreds of pages.

This research focuses on summarizing information retrieved from various online sources on the same topic, which is structured and classified by key categories and specifications. The methodology used combines both extractive mechanisms, which are logically closer to the work of human experts, and abstractive mechanisms, which are built upon a linguistic model of the domain area developed in previous work by the author.

Abstractive summarization is a type of natural language processing (NLP) that combines technologies such as machine learning, data mining, semantic representation, neural networks, natural language analysis and synthesis, etc., to generate a substantiated, meaningful, and coherent summary. The abstractive approach represents a form of abstraction beyond standard methods for searching and extracting grammatical patterns, whereas extractive summarization involves selecting and combining the most significant sentences and phrases from the source documents [1].

The objective of this research is to automate the processes of generating condensed summaries of content retrieved from different web pages on the same topic. To achieve this goal, the following subtasks must be addressed:

- Automatic collection and processing of information based on specific criteria from a variety of online sources related to project funding, initiated by different organizations—startups, sustainable companies, municipalities, government bodies, schools, NGOs, etc.
- Automatic generation of summaries from the collected information, which typically consists of detailed descriptions of open funding programs, including specific conditions, deadlines, budgets, and more.

The automated AI software tool developed through the proposed methodology is designed to crawl and read project funding information from various web documents and to select, process, and prepare a condensed version containing only the most important key information for its clients.

2. Overview of Summarization Methods

The information content when generating a summary from various text documents, online sources, NoSQL databases, etc., depends on the user's needs. A good summary should cover the most important information of the original document or a cluster of documents while being coherent, non-redundant, and grammatically readable. Numerous approaches for automatic summarization have been developed to date [2].

Summary generators can be classified according to the type of primary/input document. If the input document has little text content, the extraction is from a single document. If the information content is complex, voluminous, and composed of multiple documents and semantic relationships between concepts, it is a multi-document extraction. The two types of summarization differ in the number and type of pre-natural language processing and the post-generation models used.

Topic-oriented summaries focus on the topic of interest to the user and extract information related to the specified topic. On the other hand, generic summaries try to cover as much of the informational content as possible, maintaining the general thematic structure of the original text [3].

According to the purposes they serve, automatic document summary generators can be classified as generic when the model is not trained and has no prior preparation, classification rules, or specifications on how to handle the task, and domain-specific when the generator is trained with linguistic resources, facts, and domain-specific rules to better extract and understand the problem.

There is also a query-based approach, where the summary contains only known answers to natural language questions regarding the input text [4]. The specific summary depends on the audience for whom it is intended, according to [5]. It responds to a query triggered by users. According to Patel, the user-focused technique generates a summary according to the specific interest of a user, while a summary on a specific topic pays special attention to that particular topic.

The position of sentences in the text is also considered indicative of the relevance of the sentence. For example, in news articles, the first paragraph typically contains the main information about the event, while the rest of the text provides details. In scientific papers, the most important information is usually communicated in the conclusion, so the strategy should be adapted to the type of text to be summarized, according to [6].

Some authors follow a summary generation approach by selecting sentences from the text that users of their websites copy and use for their purposes [7]. These are most often sentences copied from the introduction, conclusion, definitions in the text, etc. In the Crowd-Copy Summarizer tool proposed by the authors, the selection is made by studying how often users copy certain sentences to their clipboards. Their innovative approach combines summary generation with collective human wisdom while saving time, resources, and manual work.

The object-based approach used in [8] is based on a framework for measuring local coherence and is a representation of discourse with a network of objects that captures patterns of object distribution in the text. The proposed algorithm automatically abstracts the text into a set of object transition sequences and records distributive, syntactic, and referential information for discourse objects. The object-based approach is highly suitable for ranking-based summarization and text classification tasks.

Authors in [9] developed a learnable approach for document summarization. They use the Naive Bayes technique and compare the generated summaries with expert reference summaries. During the evaluation, they found that their model could detect 84% of the sentences from expert-written summaries [9].

The study [10] performed abstract summarization using a CNN-based seq2seq model. Furthermore, they found that their model performs effectively compared to the RNN-based seq2seq model. They applied the model to two datasets: the DUC corpus and the GigaWord corpus. After applying the model to the datasets, they obtained a RougeL score of 35% for the GigaWord corpus and 26% for the DUC corpus, concluding that their model performs better than RNN and GAN models.

Word embedding represents a continuous vector representation capable of capturing the syntactic and semantic information inherent in a given word. Several methods exist for creating word embeddings, following the distribution hypothesis—continuous bag-of-words and skip-gram. These models learn vector representations for each word using a neural network-based language model and can be efficiently trained on billions of words [11].

There are two other approaches for summarizing long texts—extractive and abstractive summarization. The extractive summarization method selects sentences from documents and ranks them based on their weight to generate a summary. The abstractive summarization method focuses on the main concepts of the document and then expresses these concepts in natural language [12]. Extractive summarization creates summaries by selecting a subset of sentences from the original document, contrasting with abstractive summarization, where the information in the text is rephrased [3].

The authors in [3] build on the well-known centroid-based summarization by proposing a graph representation of a document cluster, where nodes represent sentences, and edges are defined based on the similarity relationship between pairs of sentences in the text. They introduce three new centrality measures based on the concept of “prestige” in social networks [3]. The centroid of a group of documents in [3] is defined as a pseudo-document consisting of words that have results above a predefined threshold, formed by the word frequency in the cluster multiplied by a value calculated for a much larger and similar dataset. According to the authors, in centroid-based summarization, sentences that contain

more words from the centroid of the cluster are considered central. The authors' approach was applied in the first web-based multi-document summarization system [13].

According to [14], modern neural network-based approaches have achieved promising results thanks to the availability of large-scale datasets containing hundreds of thousands of document-summary pairs. The authors work on a type of centroid-based summarization by considering a popular graph-based ranking algorithm and how the centrality of a node (sentence) is calculated in two ways: (a) they use a BERT model to train a neural representation for sentence capture and (b) they build graphs with directed edges and claim that the contribution of any two nodes to their respective centrality is influenced by their relative position in a document.

There is also a controlled abstractive technique for generating summaries from words and sentences that do not appear in the input text [4]. The technique consists of training a controlled learning model with a dataset containing articles and their summaries.

Multiple strategies are used to organize information in multi-document summarization to ensure that the summary generated is coherent. There is a difference between generating a summary of just one document and summarizing all the key information from different primary sources. One of the necessary conditions for success is the study of the properties of structured information for a specific domain area and introducing constraints based on the chronological order of events and thematic/semantic coherence of the information [15].

An approach that uses hierarchical structure and ontological links to ensure accurate measurement of term similarity and improve summary quality is proposed by the scholar Alami [16]. His method is based on a two-dimensional graph model that uses both statistical and semantic similarities. The similarity is based on the content overlap between two sentences, while semantic similarity is calculated using semantic information extracted from a lexical database, which allows their system to apply reasoning by measuring the semantic distance between real human concepts [16].

There are two existing strategies for applying pre-trained language representations to downstream tasks: feature-based and fine-tuning. The feature-based approach, such as ELMo [17], uses task-specific architectures that include pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer OpenAI GPT [18], introduces minimal task-specific parameters and is trained on downstream tasks by fine-tuning all pre-trained parameters. Both approaches share the same goal during pre-training, where they use unidirectional language models to learn general language representations [19].

So-called algebraic methods include a logarithmic strategy for classifying sentences using a naive Bayes classifier to decide whether a given sentence should be included in the summary [20]. The authors claim that their approach reduces the source text by up to 83%. Their method consists of three stages: (i) natural language processing—lemmatization, normalization, and removal of stop words; (ii) keyword extraction based on their frequency in the text, and (iii) summary generation.

Summary generation systems can be controlled or uncontrolled, depending on whether there is a training dataset available or not. A controlled system is trained using a collection of labeled metadata. Uncontrolled systems do not rely on training instances created by human experts. In terms of input documents' language, automatic summarization tools can be monolingual or multilingual [5].

Statistical approaches to content selection rely on calculating the frequency of relevant keywords in the primary document, as mentioned by Saggion in [6]. The main mechanism is to measure the frequency of each word in a document and then adjust it with additional data, such as the inverse document frequency of the word in a general document collection. According to Saggion, this increases the score for words that are not too common

and reduces the score for frequently occurring words. Statistical methods assume that the relevance of a specific concept in a given text is proportional to its frequency in the document. Once the keywords are identified, sentences containing them can be selected using various mechanisms for evaluating and ranking sentences (e.g., the relevance of a sentence may be proportional to the number of keywords it contains) [6].

3. Methodology

The selection of an appropriate methodology for data extraction is crucial to ensure the efficiency of the model. The proposed approach for summary generation is abstractive and is based on the evaluation of key lexical features and the extraction of important and useful information to be included in the summary. Controlled extractive strategies have been employed, such as defining structural rules embedded in the text, grammatical logical models, and semantically based methods for natural language processing and document summarization. A pre-trained domain-specific linguistic model was also used to classify key specifications.

In our previous research, we specialized in extracting specific grammatical categories annotated to lexical units in the text. For the pre-text processing, a controlled linguistic model was used for the relevant domain, developed in previous studies by the author. Similarly to the WordNet package, this model stores the words and phrases from the linguistic model, enriched with various types of metadata, including synonym sets that cover the basic concepts forming the relationships in the linguistic database [21–23].

In a previous study, the author created a Conceptual Model of a Domain Area (CMDA), which contains linguistic concepts paired with named objects from the system database, in cases of information retrieval in question–answer systems. For this purpose, property categories (TableN, ColumnN, Aggr_fun, Value, etc.) are selected that correspond to the role and place occupied by a given object in the database of a given information system (Table 1) [22].

Table 1. Description of elements from the database.

Property	Description
TableN	Table name
ColumnN	Column name
DefaultColumn	The default column in the table
Aggr_fun	Aggregate function
Function	Specific function
Value	Value
isColumnOf	Relation between ColumnN and TableN
relatesColToTable	Relation between two tables
DataType	The data type of each attribute

Thus, the CMDA model sets the skeleton of the linguistic database, defining a hierarchy of classes and properties, as well as the directions of inheritance and finding connections between objects (Table 2). The CMDA is filled with standard terminology from the modern natural language and allows integration, expansion, and updating of data while avoiding duplication. It stores information about domain concepts, possibly duplicating them (Table 2), due to the fact that: (a) the same concept or phrase may be present as an entry in several tables; (b) several tables may contain a field with the same name [22].

Table 2. Conceptual Model of the Domain Area (CMDA).

id	Phrase	db_Element	Property	Parent_id	Data_Type
1	Concept 1	table1	TableN	-	Text
2	Concept 2	table1.column2	ColumnN	1	Text
3	Concept 3	table1.column4	ColumnN	1	Text
5	Concept 5	table2.column2 table1.column2	Value Value	3 1	Number Number
6	Concept 6	count	Aggr_fun	-	Number
7	Concept 7	min	Aggr_fun	-	Number
..	..	..	..	..	..
n	Concept n	table2.column1 table3.column4	DefaultColumn DefaultColumn	3 ..	Number Text

The author then extends the model by adding a semantic and lexical declarative description of each concept in the CMDA, as well as its place in the extensive text for subsequent generalization processes. In the obtained linguistic model, the semantic units (words, phrases) interact through stable relations with each other. The feature tables that linguistically describe the concepts at this stage are based on the properties of the language units. The language units are annotated with a syntactic category, part of speech, lexical meaning, synonyms and equivalent values, hypernyms, hyponyms, etc. (Table 3). Linguistic annotation data are presented in the form of key-value pairs. Each category receives a single value derived from predefined sets [21].

Table 3. A set of attributes describing each concept in a model of a domain area.

Metadata	
Concept	Token_id
	Synsets
	Description/Lexical meaning
	Part of speech/N, Adj, Adv, S, /
	Semantic class/category, role/
	Core
	Basic form/Lemma
	Hypernym
	Hyponym
	Belongs to/Related to

Synsets in the linguistic model can be filled in automatically, with open-access tools available, or manually with the help of linguistic experts [22]. The conceptual and linguistic model of a domain area includes concepts related to the area. This makes the models universal for use in different information systems. Naturally, each system has its specifics, and then the models can be reused without being redeveloped [23].

The author believes that a fully data-driven approach is guaranteed to generate deterministic segmentation for every possible sequence of characters and has also created a sentence model for future summarization.

The sentences that are later included in the summary are determined by the syntactic metadata and semantic annotations they contain (Table 4). The importance of the sentences depends on the roles of the tokens contained in them and is defined by a sentence

importance coefficient, class, and cosine similarity. Ultimately, the summarized textual information is a subset of the primary documents. In this way, the initial extensive information is reduced to the most essential parts relevant to the searched topic. It is necessary to predefine the key features, variables, constraints, roles, categories, grammatical rules, and models in the text.

Table 4. Description of sentences.

Sentence	Attributes
Sentence	Pattern_id
	Type/short, long/
	Part of text/title, main part, conclusion, abstract/
	Contains numbers
	Label class/neutral, logic output, contradiction, positive, negative/
	Cosine similarity
	Importance coefficient

The process of generating summarized content follows the sequence below:

Step 1: Automatic crawling of multiple online information sources—the financial AI tool receives primary sources of information—URL addresses of web pages that are searched for key information using certain search criteria.

Step 2: Generation of summarized content—processes are initiated to search for linguistic patterns in the documents, and explicit and implicit parameters in the models, which will shape and be included in the shortened version of the document.

Algorithm for executing Step 1—automatic crawling of multiple different online sources (Figure 1):

- Collecting various primary online information sources—URLs known to publish the most recent information on launched procedures and funding programs for projects. Preliminary preparation of online sources.
- Analyzing the structure of web documents to identify the required information and determining specific data fields to be extracted.
- Extracting precisely defined key data (web scraping) directly from the DOM tree of the web documents. For this purpose, various methods are applied, including analysis, comparison with models, and rule-based or axiom-based extraction.
- Web crawling online content to find or discover subpages or URL pages linked within it. Web crawling is used for content indexing.
- Applying reliable data validation mechanisms to ensure the accuracy and integrity of the data.
- The data extraction process is designed to handle large volumes of documents.

For crawling online sources and collecting their informational content, two additional techniques were employed—web scraping and web crawling. Their combined use within specially designed Python (used version Python 3.10) functions supports the automatic navigation, reading, and verification of numerous web documents without the need for expert human intervention. Web scraping and web crawling automate manual tasks, reduce the time and cost of document processing, and enable the sequential and efficient handling of large volumes of documents. They are used to ensure higher data accuracy, as they minimize the possibility of human errors and data inconsistencies.

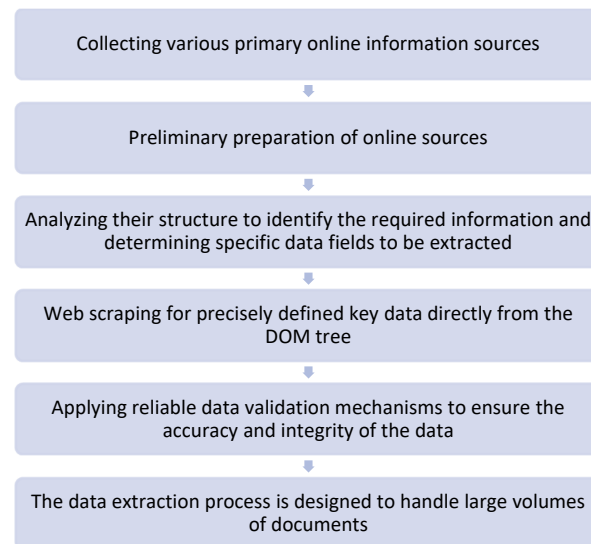


Figure 1. Algorithm for executing Step 1.

Automated data extraction can be configured and trained to work with different types and models of documents. Once the necessary information has been collected from various online sources, the procedure for generating the shortened, summarized content (summary) involves the extraction of textual fragments with key features (highlights) that will be included in the summary.

Algorithm for executing Step 2—summary generation (Figure 2):

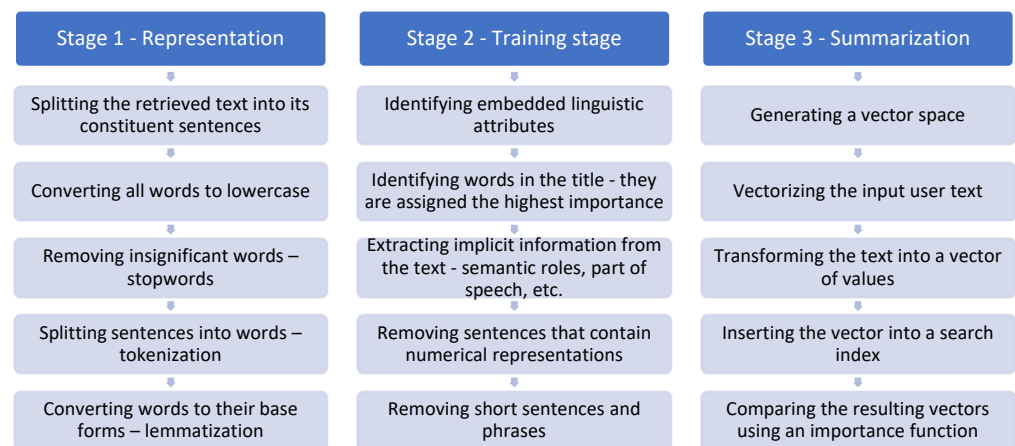


Figure 2. Algorithm for executing Step 2.

Stage 1: Representation

- Splitting the retrieved text into its constituent sentences.
- Converting all words to lowercase.
- Removing insignificant words—stop words, punctuation, special characters, and white spaces.
- Tokenization—splitting sentences into their constituent linguistic units.
- Normalization (lemmatization)—converting words to their base forms.

Stage 2: Education

- Identifying embedded linguistic attributes—words present in the title of the text are assigned the highest importance/weight.
- Extracting implicit information (semantic roles, part of speech) from the text.

- Removing sentences that contain numerical representations. Numbers do not contribute to the summarization process.
- Removing short sentences and phrases. These are considered insufficiently descriptive, convey limited information, and do not support the formulation of conclusions.

Stage 3: Summarizing

- Generating a vector space.
- Vectorizing the input user text.
- Transforming the text into a vector of values.
- Inserting the vector into a search index.
- Comparing the resulting vectors using an importance function.

In the Representation stage, general knowledge about the domain is collected, a linguistic model is constructed, and the necessary specific metadata for the functional units involved (the objects forming the model) is gathered. Known declarative knowledge (facts, grammatical characteristics, constraints, relations) is collected and structured to make the data accessible and understandable for subsequent processing. Similarly to phrases in natural language, the model for describing a domain has a complex structure and is constructed by systematizing simple metadata about objects, their characteristics, properties, criteria, constraints, and rules. Taking into account the quantity of metadata, new semantic representations are constructed based on compositions of symbolic expressions and analogies between syntactic structures and semantics. Knowledge continues to be collected and accumulated in the next stage—the training process (Figure 2).

An increase in performance in automated summary generation is achieved over time and is related to the improvement of the linguistic resource and the enhancement of the tools that validate and build it. Training and representation are processes that run through multiple iterations and corrections. Knowledge about the domain is accumulated during the training process.

In the Training stage, the procedural representation of knowledge is formed. The collected knowledge is embedded into procedures that function independently of the meaning of the knowledge itself. At this point, the new free functional parameters are finely tuned through the environment previously modeled in the Representation stage. In this stage, the system is trained to extract explicit and implicit information from the original documents. Mechanisms are defined for managing and controlling operations used to solve specific subtasks. Solving a specific task is reduced to a search task, during which statistics, rules, data, and control mechanisms for interaction are used. Rules operate on the data domain, while control interactions are determined by the rules. Statistical processing identifies the documents where the sought-after data is most likely to be found. Both sequential processing with the accumulated knowledge and parallel distributed processing are performed. Sometimes the accumulated knowledge is incomplete and inaccurate, and probabilistic reasoning must be applied. In such cases, it is said that the summarization generation mechanism operates under conditions of uncertainty. An important aspect is the capturing of exceptions, errors, and monitoring feedback signals from the system's functional elements. The feedback mechanism can verify hypotheses and revise and improve them if necessary.

In the Summarization stage, the process of automatic summarization begins with the vectorization of the user's input text. Using a multidimensional orthogonal space model, all sentences in the text are transformed into strings or radius vectors of values $X = (x_1, x_2, \dots, x_n)$ as a sample from a multivariate variable in n -dimensional Euclidean space, and their weight is calculated. The actual significant data in this case are the most frequently occurring words in the text, and when they are identified within a sentence, this increases

the sentence's importance score. Through better management of the weighting function and coefficients to determine the weight, the error in identifying the most important result can be controlled—something not possible in some other models.

4. Conclusions

Summary generation is a technology for identifying the most important information and simultaneously solves several problems—it helps users quickly find the information they are looking for, reduces the size of the original text document, and briefly presents its main idea.

The study presents a review of existing approaches to text summarization and a hybrid approach for the automatic generation of condensed summaries extracted from online sources for project funding, combining both extractive and abstractive mechanisms. A methodology is proposed that includes: (i) a linguistic model of the domain, consisting of concepts annotated with their linguistic characteristics, such as semantic role/category, synonym set, part of speech, etc., (abstraction), required for preliminary NLP; (ii) a process model for preliminary and core text processing, review and scraping of informative web content (extraction); and (iii) summarization of the extracted textual content.

The focus of the research is on creating a methodology for developing an automated tool for searching a large set of online sources and generating a summary of large-volume documents that meet the specified search criteria. The research aims to develop a model of process for creating a financial assistant for collecting and processing up-to-date information from a wide range of sources on project funding initiated by various organizations—start-ups, sustainable companies, municipalities, government agencies, schools, NGOs, and others.

The software tool created using the proposed methodology has the freedom to be developed in any programming language. In future developments, the author will also present the software Python (version 3.10) implementation of the completed tool, created using the current methodology, supported by experimental data for extracting information from various online sources and generating a summary from extensive text documents. We will prove that the developed models and methodologies are reliable only when we present the finished software tool. Our aspiration is for it to be accessible, adequate, and sustainable. Experimental results will show its practical applicability and thus contribute to the proposed methodology.

The input information for the available open calls for funding is available on the Internet in multiple languages (we are interested in English and Bulgarian), while the output information intended for users is in Bulgarian. This contributes to the usefulness of the service—localization, presentation in the native language, and adaptation of the information to be understandable for non-experts. At a future stage, the service is expected to be prepared and offered in other languages (scaling to other markets). An analysis of current studies shows that no similar tool exists in Bulgaria, and the developed methodology can be transferred and applied in other domain areas.

The proposed hybrid approach and methodology for creating an AI tool capable of extracting the most important semantically related information from large volumes of data and generating summarized content can also be applied in areas outside finance. Such a tool is needed and effective in education, healthcare, and the technical and economic sciences. It contributes to research progress, optimizes resources, and saves experts' time by automating logical analysis and comprehension of large datasets. In conclusion, since technologies and techniques for automatic summarization continue to evolve, and the specific requirements of users continue to grow, it is logical for applications, including the proposed methodology and AI summarization tool, to continue to be improved and developed.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The author would like to thank the University of Food Technologies for the financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yadav, A.K.; Ranvijay Yadav, R.S.; Maurya, A.K. State-of-the-art approach to extractive text summarization: A comprehensive review. *Multimed. Tools Appl.* **2023**, *82*, 29135–29197. [\[CrossRef\]](#)
2. Yao, J.; Wan, X.; Xiao, J. Recent advances in document summarization. *Knowl. Inf. Syst.* **2017**, *53*, 297–336. [\[CrossRef\]](#)
3. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [\[CrossRef\]](#)
4. Awasthi, I.; Gupta, K.; Bhogal, P.S.; Anand, S.S.; Soni, P.K. Natural Language Processing (NLP) based Text Summarization—A Survey. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 1310–1317. [\[CrossRef\]](#)
5. Patel, D.; Shah, S.; Chhinkaniwala, H. Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. *Expert Syst. Appl.* **2019**, *134*, 167–177. [\[CrossRef\]](#)
6. Saggion, H.; Poibeau, T. Automatic Text Summarization: Past, Present and Future. In *Theory and Applications of Natural Language Processing*; Springer: Berlin/Heidelberg, Germany, 2013. [\[CrossRef\]](#)
7. Kirsh, I.; Joy, M. An HCI Approach to Extractive Text Summarization: Selecting Key Sentences Based on User Copy Operations. HCII 2020. In *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2020; Volume 1293. [\[CrossRef\]](#)
8. Barzilay, R.; Lapata, M. Modeling Local Coherence: An Entity-Based Approach. *Comput. Linguist.* **2008**, *34*, 1–34. [\[CrossRef\]](#)
9. Hanif, U. Research Paper Summarization Using Text-To-Text Transfer Transformer (T5) Model. Doctoral Dissertation, College of Ireland, Dublin, Ireland, 2023.
10. Zhang, Y.; Li, D.; Wang, Y.; Fang, Y.; Xiao, W. Abstract Text Summarization with a Convolutional Seq2seq Model. *Appl. Sci.* **2019**, *9*, 1665. [\[CrossRef\]](#)
11. Rossiello, G.; Basile, P.; Semeraro, G. Centroid-based Text Summarization through Compositionality of Word Embeddings. In Proceedings of the Multiling 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres, Valencia, Spain, 3 April 2017.
12. Sajjan, R.S.; Shinde, M.G. A Detail Survey on Automatic Text Summarization. *Int. J. Comput. Sci. Eng.* **2019**, *7*, 991–998. [\[CrossRef\]](#)
13. Radev, D.; Blair-Goldensohn, S.; Zhang, Z. Experiments in single and multidocument summarization using MEAD. In Proceedings of the First Document Understanding Conference, New Orleans, LA, USA, 9–13 September 2001.
14. Zheng, H.; Lapata, M. Sentence Centrality Revisited for Unsupervised Summarization. *arXiv* **2019**, arXiv:1906.03508.
15. Barzilay, R.; Elhadad, N. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Intell. Res.* **2002**, *17*, 35–55. [\[CrossRef\]](#)
16. Alami, N.; Mallahi, M.E.; Amakdouf, H.; Qjidaa, H. Hybrid method for text summarization based on statistical and semantic treatment. *Multimed. Tools Appl.* **2021**, *80*, 19567–19600. [\[CrossRef\]](#)
17. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365. Available online: <https://arxiv.org/abs/1802.05365> (accessed on 5 May 2025).
18. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding with Unsupervised Learning*; Technical Report; OpenAI: San Francisco, CA, USA, 2018.
19. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
20. Sreenivasulu, G.; Thulasi Chitra, N.; Sujatha, B.; Venu Madhav, K. Text Summarization Using Natural Language Processing. In *Lecture Notes in Networks and Systems*; Springer: Singapore, 2022; Volume 191. [\[CrossRef\]](#)
21. Zhekova, M.; Totkov, G. Model of process and model of natural language processing system. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *878*, 012028. [\[CrossRef\]](#)

22. Zhekova, M.; Totkov, G.; Pashev, G. Methodology for creating natural language interfaces to information systems in a specific domain area. In Proceedings of the 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20–22 July 2022; pp. 1–6. [[CrossRef](#)]
23. Zhekova, M.; Totkov, G. Language models and algorithm for natural language text recognition. *J. Inform. Innov. Technol. (JIIT)* **2022**, *5*, 29–36.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.