Network Text Analysis Framework for Mapping Research Trends: A Neural Architecture Search Case Study

Petra Vidnerová[1,*,†], Roman Neruda[1,†]

[1]*The Czech Academy of Sciences, Institute of Computer Science*

Abstract

Understanding the evolution of research trends is critical for navigating rapidly developing scientific literature. Large Language Models (LLMs) offer powerful tools for analysing scientific texts, enabling the extraction of key concepts and the construction of semantic networks. These capabilities can support the study of emerging ideas and research trends through graph-based representations.

In this paper, we present a network-based text analysis framework designed to map the evolution of scientific knowledge. Our goal is to extract conceptual structures from research papers and construct graphs that represent both the occurrence of terms and their interrelationships. The integration of temporal information allows us to track the emergence and transformation of research themes.

We demonstrate this framework using a case study on Neural Architecture Search (NAS) field, a fast-growing subfield in machine learning focused on the automated design of neural networks. Using data from ArXiv combined with metadata and citation records from OpenAlex, we construct and analyse graphs of keywords and articles. This allows us to reveal the dynamics of the NAS research landscape and highlight methodological trends and conceptual shifts.

Keywords

network text analysis, semantic graphs, large language models application, neural architecture search

1. Introduction

Text analysis has become an essential tool for systematically exploring large volumes of scientific literature, enabling researchers to uncover patterns, trends, and relationships that would be difficult to detect through manual review. Techniques such as keyword extraction, topic modeling, and network analysis allow for the identification of emerging research themes, influential publications, and the conceptual organization of knowledge across expansive academic corpora. This is particularly valuable in fast-growing fields, where the volume of new publications makes traditional literature review methods increasingly impractical. By offering scalable and objective insights, text analysis supports tasks such as research mapping, literature synthesis, and the discovery of gaps or future research opportunities.

Traditional approaches [1] to network text analysis often depend on manually curated keywords, rule-based extraction, or shallow natural language processing methods. These can be labour-intensive, prone to inconsistency, and limited in their ability to capture nuanced meanings and implicit relationships. In contrast, large language models (LLMs) [2] offer a more powerful and flexible alternative. They enable context-aware, automated extraction of key concepts and connections, handling linguistic variation and ambiguity with greater ease. This not only streamlines the analytical process but also enhances the depth and accuracy of the resulting networks, making LLMs a transformative tool in the modernization of text analysis.

In this work, we propose a methodological framework for text analysis designed to systematically explore and extract meaningful patterns from textual data. Our approach integrates structured techniques to identify key concepts, relationships, and trends within a corpus, enabling a deeper understanding of

ITAT'25: Information Technologies – Applications and Theory, September 26–30, 2025, Telgárt, Slovakia

*Corresponding author.

†These authors contributed equally.

petra@cs.cas.cz (P. Vidnerová); roman@cs.cas.cz (R. Neruda)

https://www.cs.cas.cz/~petra (P. Vidnerová)

0000-0003-3879-3459 (P. Vidnerová); 0000-0003-2364-5357 (R. Neruda)

the underlying content. By combining both qualitative insights and quantitative metrics, the framework offers a comprehensive tool for analysing large volumes of text, making it applicable across a range of domains where text-based information plays a central role.

The designed paper networks enable better orientation among scientific papers as well as retrieval mechanisms delivering papers based on topic similarity.

2. Methodology

In this section, we propose a methodology for analysing a collection of research papers using network text analysis. We begin by outlining the process of constructing the dataset, which includes assembling the papers and incorporating citation information. Next, we detail the procedure for selecting keywords and generating network graphs.

Dataset

We construct a dataset of research papers using the arXiv repository. Specifically, we use the arXiv API to download N papers based on a search query related to the topic or field of interest.

In our case study, we retrieved 10,000 papers using the query "neural architecture search." For each paper, we obtained both the PDF and metadata, including the title, authors, abstract, and other relevant information.

However, not all papers retrieved through this method are truly relevant to the target topic—particularly when using a large value of N. To address this, we apply a filtering step using an LLM. For each paper, we provide the abstract and a prompt asking whether the content is related to the target topic.

After filtering out irrelevant papers, we retained a final set of 2,423 arXiv papers that we believe to be related to Neural Architecture Search.

Citations

The second step involves retrieving citation information from the OpenAlex [3] database. The simplest approach is to locate each paper in OpenAlex using its arXiv ID and extract its citation data. However, many papers also have peer-reviewed versions published in conferences or journals, which are often cited more frequently than their arXiv preprints.

To account for this, we search OpenAlex for all papers with identical titles (ignoring differences in punctuation and whitespace) and aggregate the citations from these matched entries. We then consider only those citations that originate from the set of arXiv papers compiled in the previous step.

Keywords

To enable the analysis of topic trends, we associate each paper with a set of keywords. We consider two approaches for keyword generation. The first involves prompting a large language model (LLM) with a paper's abstract and directly requesting a list of keywords. The second approach starts with a manually curated set of fixed keywords; for each abstract-keyword pair, the LLM is then asked whether the keyword is relevant to the given abstract.

The first approach can yield rich and diverse keywords, but it requires substantial postprocessing to unify synonyms and eliminate overly general terms. The second approach produces a clean, boolean keyword vector for each paper, which is easier to analyse, but it depends heavily on the quality and scope of the initial keyword set, which may be limited or biased.

In this study, we adopt a hybrid method. We begin by generating a set of keywords for each paper using the first approach. We then aggregate all generated keywords, select the most frequent ones, and filter out overly general terms (e.g., "neural network" in the context of neural architecture search). The resulting refined list serves as our fixed keyword set for further analysis.

Networks

A key advantage of network text analysis is its ability to reveal structural relationships between concepts, keywords, or entities within a large text corpus. This is achieved by constructing graphs—sets of vertices V and edges E, where each edge is a pair (u, v) such that $u \in V$ and $v \in V$. Vertices represent entities (e.g., keywords or papers), and edges represent the relationships between them.

In our study, we construct three types of networks: a keyword network, a paper network, and a citation network.

- In the keyword network, vertices correspond to individual keywords. An edge is created between two keywords if they co-occur in at least one paper. Each edge is weighted by the number of shared papers in which the two keywords appear.
- In the paper network, vertices represent papers, and an edge is established between two papers if they share one or more keywords. The edge weight reflects the number of shared keywords.
- In the citation network, vertices again represent papers, but edges are directed: an edge (u, v) indicates that paper u cites paper v.

These networks serve both as analytical tools and visualisation aids, helping to map relationships and navigate large collections of research papers. Additionally, we can compute various network properties (such as centrality measures [4]) to capture the broader structure of the network. These metrics help identify key nodes based on their position within the overall graph topology.

The first centrality measure we use is the *PageRank* [5], which evaluates the importance of a document based on both the number and the quality of links pointing to it. In our context, the *PageRank* centrality helps identify influential papers.

Secondly, we will use the *betweenness* centrality [6], a graph-based metric that reflects how often a node appears on the shortest paths between other nodes. This measure highlights nodes that serve as bridges within the network, such as interdisciplinary papers. In the context of a single research field, high betweenness may indicate papers that connect distinct concepts or subtopics.

3. Neural Architecture Search

As a case study, we chose the field of Neural Architecture Search (NAS) to showcase the capabilities of the proposed text analysis method.

NAS is a subfield of automated machine learning (*AutoML*) focused on the automatic design of neural network architectures. Traditional deep learning models require significant manual effort and expert knowledge to design effective architectures, which may not generalise well across tasks or datasets. NAS aims to automate this process by searching through a predefined space of possible architectures to find models that achieve optimal performance for a given task. It typically involves three main components: the search space (defining which architectures can be explored), the search strategy (how architectures are sampled or generated), and the performance estimation strategy (how the quality of each candidate architecture is evaluated) [7].

Over the past few years, NAS has seen rapid advances in both efficiency and effectiveness. Early approaches, such as those based on reinforcement learning or evolutionary algorithms, were computationally expensive, often requiring thousands of GPU hours. More recent methods—like differentiable NAS (e.g., DARTS [8]) or weight-sharing approaches—have significantly reduced the cost of the search process, making NAS more accessible and practical. NAS has been applied successfully in areas such as image classification, object detection, and natural language processing, and continues to evolve toward broader applicability, better generalisation, and improved search efficiency.

In the next section, we describe our findings on the field of NAS.

4. Results

We used a dataset of research papers retrieved from arXiv, following the procedure outlined in Section 2, resulting in a collection of 2,423 papers. As the LLM, we used the *tiger-gemma-9b-v3:fp16* model because of its relatively small size and good performance. Visualisations were created using the *graph-tool* library [9]. The code is publicly available at [10].

The temporal distribution of these papers is shown in Fig. 1. While the topic has appeared in the literature since the late 1990s, a significant increase in research activity began after 2017. This rapid

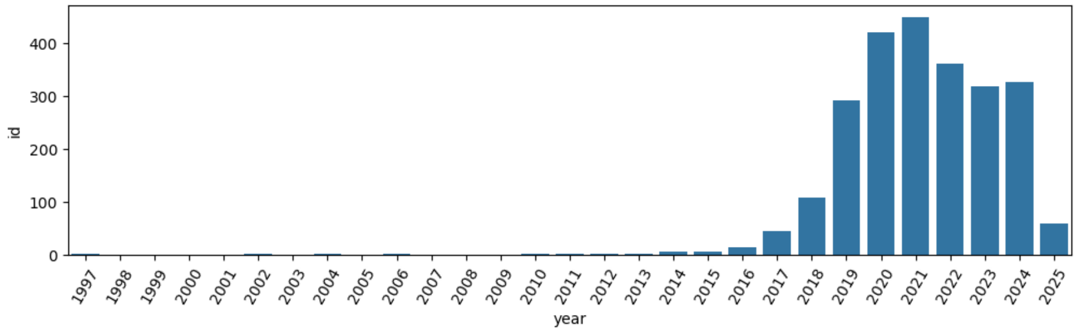

Figure 1: The numbers of NAS papers over time.

growth was driven by the increased availability of high-performance hardware and the widespread adoption of deep learning techniques.

The following figure, Fig. 2, displays the proportion of papers associated with each keyword relative to the total number of papers. Based on these keyword counts only, we can make several observations.

First, the keywords include several objectives such as efficiency, accuracy, inference time or latency, robustness, sparsity, adversarial robustness, and energy efficiency. It is evident that the proportion of papers addressing secondary but still important objectives—like energy efficiency or robustness—is much smaller compared to those focusing on accuracy or overall efficiency. Although multi-objective optimisation is increasingly important today and solutions face multiple demands, many studies continue to concentrate solely on accuracy.

Objectives such as latency and energy efficiency correspond to hardware-aware NAS, which focuses on optimising neural architectures not only for performance but also for practical deployment constraints on specific hardware platforms. In Fig. 3, we can see the proportions of keywords over time, and we can see that the importance of hardware-aware NAS increased over time.

Another group of keywords relates to the optimisation techniques used in the NAS process. Notably, multi-objective optimisation plays a significant role, appearing in nearly 30% of the papers. The most prominent optimisation methods include evolutionary algorithms, differentiable architecture search, reinforcement learning, and Bayesian optimisation.

Fig.4 shows the number of papers using different optimisation algorithms, with evolutionary algorithms and differentiable architecture search clearly dominating, and Bayesian optimisation being the least common. However, the trend over time, illustrated in Fig.5, tells a different story: differentiable architecture search methods have gradually emerged and become the dominant approach, whereas evolutionary algorithms were more prominent in the early years of the NAS field.

The next step is to move beyond individual keywords and examine the keyword graph. In Fig.6, keywords are represented as vertices, with the size of each circle proportional to the number of associated papers. The complete graph is shown in Fig.7 (left), while Fig. 7 (right) highlights the most significant edges connecting the keywords.

The strongest edges are:

- latency - inference time
- performance prediction - surrogate model
- multi-objective optimization - performance prediction
- model compression - multi-objective optimization
- hyperparameter optimization - performance prediction
- multi-objective optimization - inference time
- hardware aware - multi-objective optimization

The strong connection between latency and inference time is expected, as the two terms are essentially synonymous. Similarly, the prominent edge between performance prediction and surrogate model

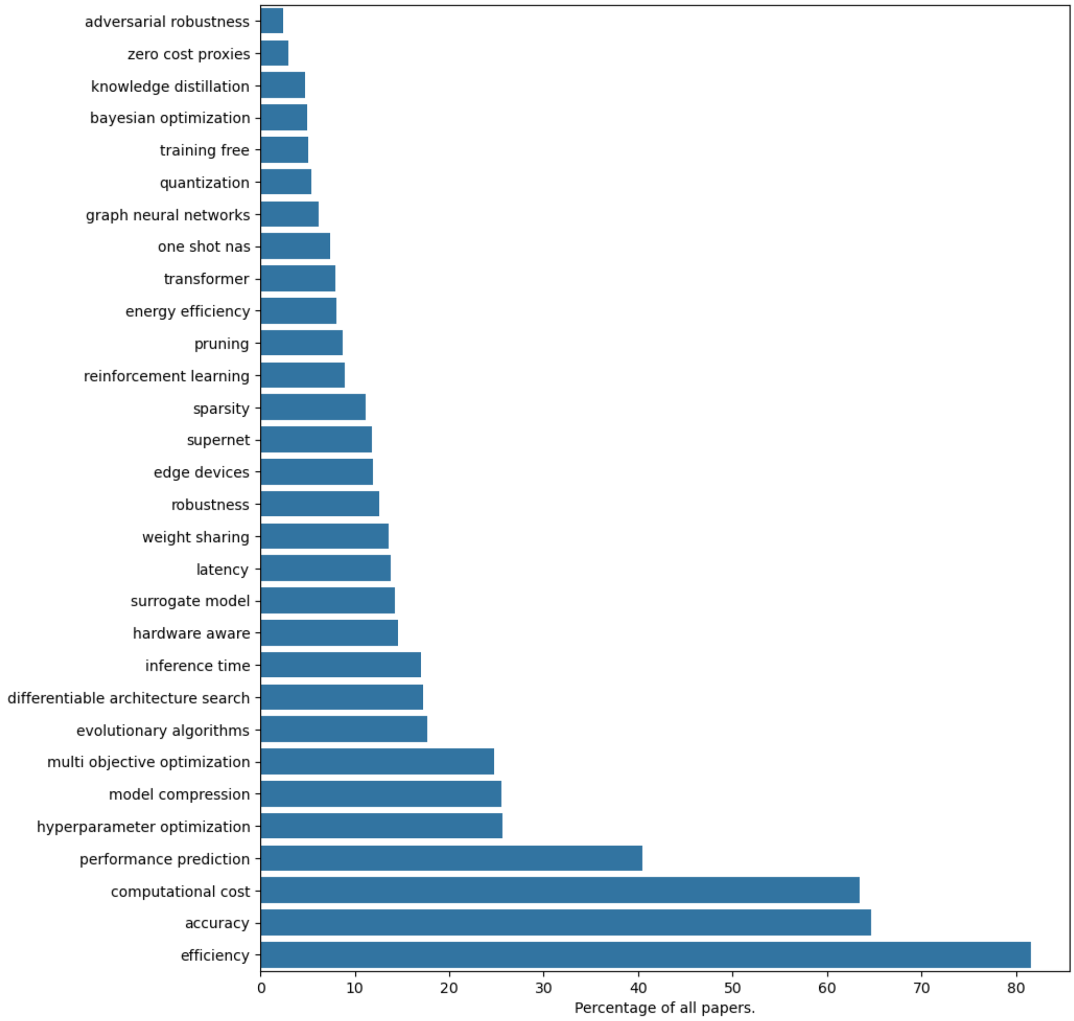

Figure 2: The pefcentange of papers according individual keywords.

highlights the central role of surrogate models in performance prediction tasks. Additionally, the strong link between hardware-aware NAS and multi-objective optimization reflects the inherently multi-objective nature of hardware-aware approaches.

A different type of graph is presented in Fig. 8, where each node represents a paper and edges indicate shared keywords. This graph includes only papers tagged with the "evolutionary algorithms" keyword, and outlier nodes have been filtered out. The resulting clusters reveal groups of papers that share common concepts, potentially aiding in the identification of related work. Papers associated with the "performance prediction" keyword are highlighted in red.

We reach a deeper level of analysis by incorporating citation data. Fig.9 (left) presents the citation graph, excluding singleton nodes, while Fig.9 (right) displays papers with more than five citations, visualized using the algorithm from [11]. Nodes with a high number of outgoing edges—those citing many other papers—are typically review or survey articles. For example, node 1192 in the figure corresponds to the paper *Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap* [12], which provides a literature review of NAS, focusing specifically on weight-sharing methods.

The three most cited papers (taking into account only citation inside the dataset) are *SNAS: Stochastic Neural Architecture Search* [13], *ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware* [14] and *NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search* [15], with equal number of citations. The [15] is the most popular benchmark for testing NAS algorithms, so not surprisingly is referred by many NAS papers.

Figure 3: The percentage of individual topics over time.

On the other hand, papers with most references are *Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap* [12] and *A Comprehensive Survey on Hardware-Aware Neural Architecture Search* [16], both survey type papers.

Citation counts correspond only to the number of incoming and outgoing edges in the citation graph. However, they do not capture the broader structure of the network. To address this limitation, we have used the two *centrality* measures – PageRank and betweenness centrality–described in section 2.

The most influential papers according PageRank are:

- *Designing Neural Network Architectures using Reinforcement Learning* [17]
- *Efficient Architecture Search by Network Transformation* [18]

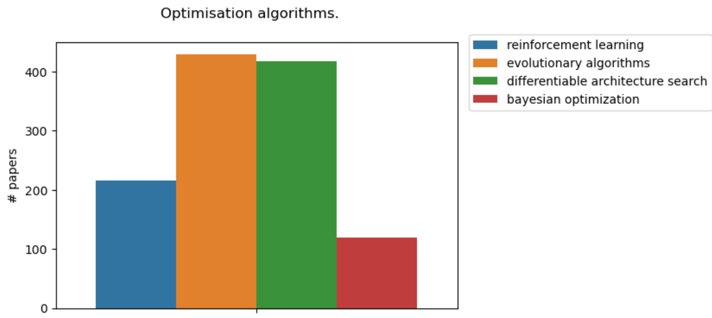

Figure 4: The numbers of NAS papers using individual optimisation algorithms.

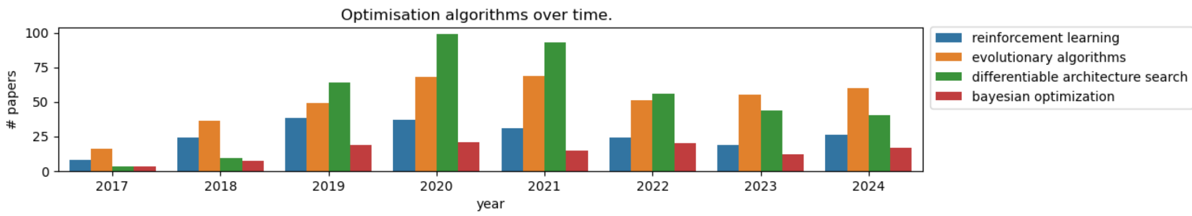

Figure 5: The numbers of NAS papers using individual optimisation algorithms over time.

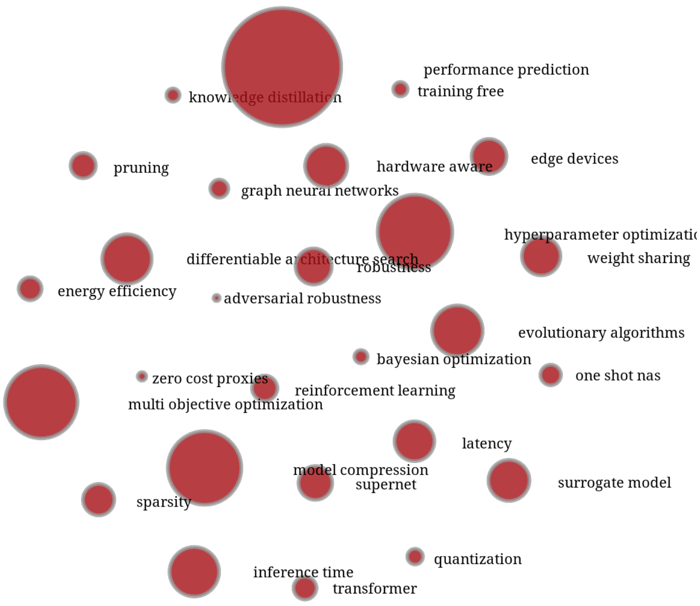

Figure 6: Keywords. Efficiency, computation cost and accuracy omitted.

- *DeepArchitect: Automatically Designing and Training Deep Architectures* [19]
- *Progressive Neural Architecture Search* [20]

The most important papers according betweenness centrality are:

- *Single Path One-Shot Neural Architecture Search with Uniform Sampling* [21]
- *Weight-Sharing Neural Architecture Search: A Battle to Shrink the Optimization Gap* [12]
- *Evaluating the Search Phase of Neural Architecture Search* [22]
- *Random Search and Reproducibility for Neural Architecture Search* [23]
- *CARS: Continuous Evolution for Efficient Neural Architecture Search* [24]

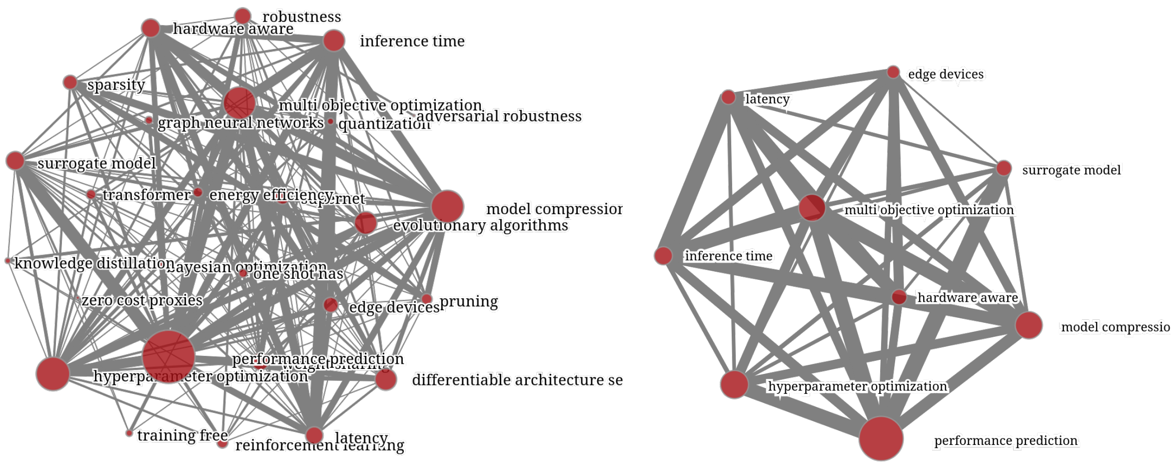

Figure 7: Keywords graph (efficiency, computation cost and accuracy omitted). (left) The complete graph. (right) The most significant edges.

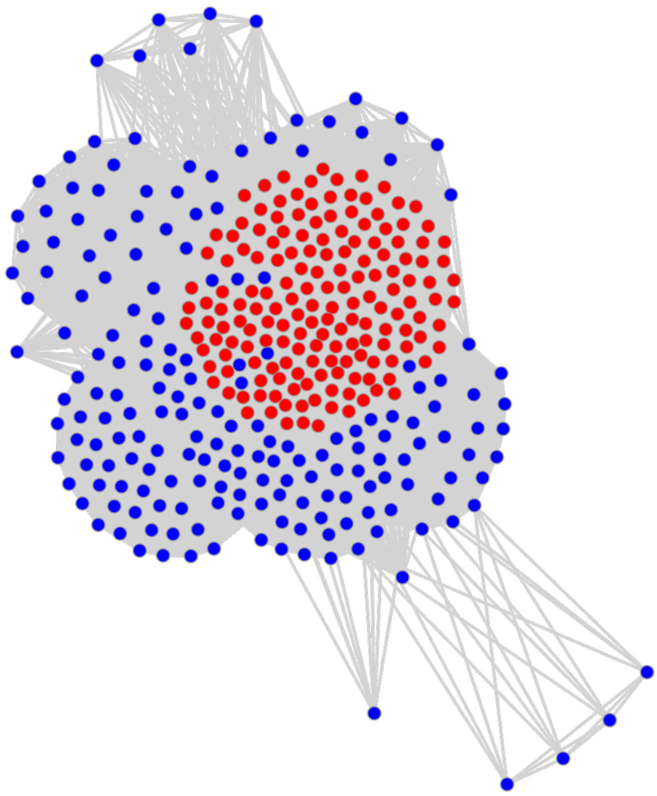

Figure 8: Graph of papers with keyword "evolutionary algorithms", papers are nodes, common keywords edges. Papers on performance prediction are highlighted by red.

- *Neural Architecture Search: A Survey* [7]

5. Conclusion

In this paper, we introduced a network-based text analysis approach supported by large language models (LLMs) to explore and better understand the structure of scientific literature. By applying our

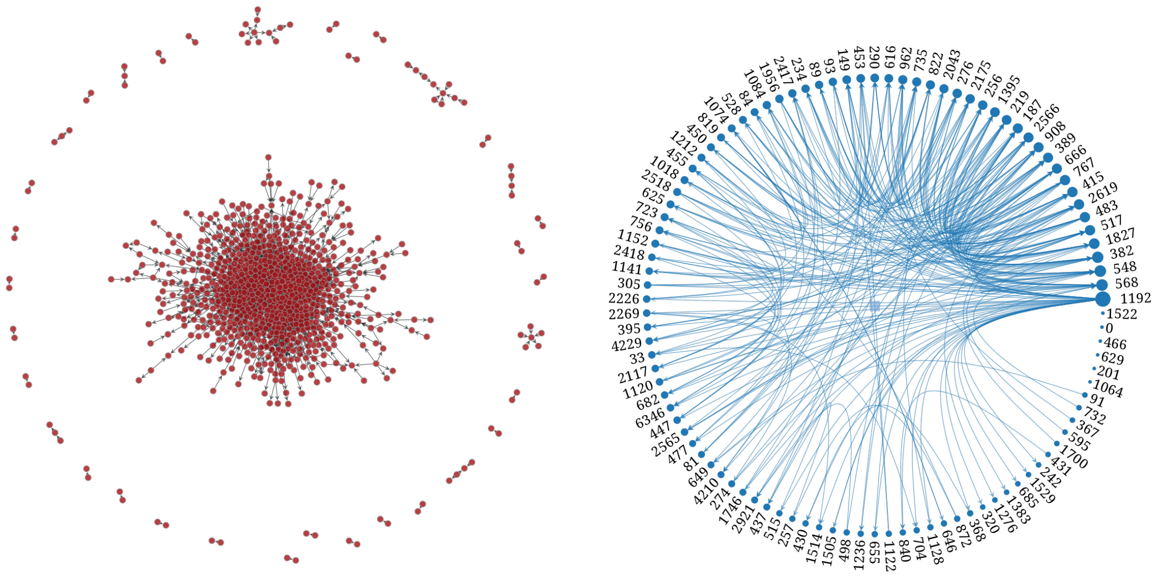

Figure 9: Visualisation of the citation graph.

methodology to the field of NAS, we demonstrated its ability to uncover key research trends, thematic clusters, and the evolution of methods over time. Through keyword co-occurrence networks, citation analysis, and centrality measures, our approach provided insights into dominant areas within the NAS domain.

The integration of LLMs enabled finer keyword extraction and classification, enhancing the quality of the resulting networks. Our hybrid method offers a scalable and interpretable way to analyse complex scientific landscapes and can be readily adapted to other research domains.

Future work will focus on extending this methodology in several directions. One promising direction is the incorporation of full-text analysis, which would allow for a deeper semantic understanding beyond abstracts and keywords. However, the work done shows that regarding keywords and topics extractions, using abstracts only gives sufficient results.

Additionally, applying dynamic network analysis could help capture how research topics and their interconnections evolve over time. Another direction involves refining the integration of LLMs—for instance, by using them not only for keyword extraction but also for automated topic labeling, trend detection, and hypothesis generation.

Future work will also explore the use of our approach for novelty detection, aiming to identify emerging topics, unconventional connections, or underexplored research directions within the scientific literature.

Acknowledgement

This work has been funded by a grant from the Programme Johannes Amos Comenius under the Ministry of Education, Youth and Sports of the Czech Republic 'Knowledge in the Age of Distrust' (reg. no. CZ.02.01.01/00/23_025/0008711) and by the long-term strategic development financing of the Institute of Computer Science (RVO:67985807).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] C. D. Manning, H. Schütze, Foundations of statistical natural language processing, MIT Press, Cambridge, MA, USA, 1999.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[3] J. Priem, H. Piwowar, R. Orr, OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, 2022. URL: https://arxiv.org/abs/2205.01833. arXiv:2205.01833.

[4] M. Newman, Networks, Oxford University Press, 2018. URL: https://doi.org/10.1093/oso/9780198805090.001.0001. doi:10.1093/oso/9780198805090.001.0001.

[5] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web., Technical Report 1999-66, Stanford InfoLab, 1999. URL: http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120.

[6] L. C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1977) 35–41. URL: http://www.jstor.org/stable/3033543.

[7] T. Elsken, J. H. Metzen, F. Hutter, Neural architecture search: A survey, 2019. URL: https://arxiv.org/abs/1808.05377. arXiv:1808.05377.

[8] H. Liu, K. Simonyan, Y. Yang, DARTS: differentiable architecture search, CoRR abs/1806.09055 (2018). URL: http://arxiv.org/abs/1806.09055. arXiv:1806.09055.

[9] T. P. Peixoto, The graph-tool Python library, figshare (2014). URL: http://figshare.com/articles/graph_tool/1164194. doi:10.6084/m9.figshare.1164194.

[10] P. Vidnerová, Source code, 2025. URL: https://github.com/PetraVidnerova/TRUST_codes.

[11] T. P. Peixoto, Hierarchical block structures and high-resolution model selection in large networks, Phys. Rev. X 4 (2014) 011047. URL: https://link.aps.org/doi/10.1103/PhysRevX.4.011047. doi:10.1103/PhysRevX.4.011047.

[12] L. Xie, X. Chen, K. Bi, L. Wei, Y. Xu, Z. Chen, L. Wang, A. Xiao, J. Chang, X. Zhang, Q. Tian, Weight-sharing neural architecture search: A battle to shrink the optimization gap, CoRR abs/2008.01475 (2020). URL: https://arxiv.org/abs/2008.01475. arXiv:2008.01475.

[13] S. Xie, H. Zheng, C. Liu, L. Lin, SNAS: stochastic neural architecture search, CoRR abs/1812.09926 (2018). URL: http://arxiv.org/abs/1812.09926. arXiv:1812.09926.

[14] H. Cai, L. Zhu, S. Han, ProxylessNAS: Direct neural architecture search on target task and hardware, CoRR abs/1812.00332 (2018). URL: http://arxiv.org/abs/1812.00332. arXiv:1812.00332.

[15] X. Dong, Y. Yang, NAS-Bench-201: Extending the scope of reproducible neural architecture search, CoRR abs/2001.00326 (2020). URL: http://arxiv.org/abs/2001.00326. arXiv:2001.00326.

[16] H. Benmeziane, K. E. Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, N. Wang, A comprehensive survey on hardware-aware neural architecture search, CoRR abs/2101.09336 (2021). URL: https://arxiv.org/abs/2101.09336. arXiv:2101.09336.

[17] B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning, CoRR abs/1611.02167 (2016). URL: http://arxiv.org/abs/1611.02167. arXiv:1611.02167.

[18] H. Cai, T. Chen, W. Zhang, Y. Yu, J. Wang, Reinforcement learning for architecture search by network transformation, CoRR abs/1707.04873 (2017). URL: http://arxiv.org/abs/1707.04873. arXiv:1707.04873.

[19] R. Negrinho, G. Gordon, Deeparchitect: Automatically designing and training deep architectures, 2017. URL: https://arxiv.org/abs/1704.08792. arXiv:1704.08792.

[20] C. Liu, B. Zoph, J. Shlens, W. Hua, L. Li, L. Fei-Fei, A. L. Yuille, J. Huang, K. Murphy, Progressive neural architecture search, CoRR abs/1712.00559 (2017). URL: http://arxiv.org/abs/1712.00559. arXiv:1712.00559.

[21] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, J. Sun, Single path one-shot neural architecture search with uniform sampling, CoRR abs/1904.00420 (2019). URL: http://arxiv.org/abs/1904.00420.

arXiv:1904.00420.
[22] C. Sciuto, K. Yu, M. Jaggi, C. Musat, M. Salzmann, Evaluating the search phase of neural architecture search, CoRR abs/1902.08142 (2019). URL: http://arxiv.org/abs/1902.08142. arXiv:1902.08142.
[23] L. Li, A. Talwalkar, Random search and reproducibility for neural architecture search, CoRR abs/1902.07638 (2019). URL: http://arxiv.org/abs/1902.07638. arXiv:1902.07638.
[24] Z. Yang, Y. Wang, X. Chen, B. Shi, C. Xu, C. Xu, Q. Tian, C. Xu, CARS: continuous evolution for efficient neural architecture search, CoRR abs/1909.04977 (2019). URL: http://arxiv.org/abs/1909.04977. arXiv:1909.04977.