

A Case Study on the Two-Sided Version of the Node of Knowledge Conceptual Framework

Martina Asenbrener Katic, Sanja Candrljic

University of Rijeka

Faculty of Informatics and Digital Technologies

Radmile Matejčić 2, 51000 Rijeka, Croatia

{masenbrener, sanjac}@inf.uniri.hr

Marina Rauker Koch

University of Applied Sciences of Rijeka

Vukovarska 58, 51000 Rijeka, Croatia

mrauker@veleri.hr

Abstract. Today, data plays a central role in supporting business decision-making, planning and operations. Most business information systems rely on relational databases to store and manage structured data. Much of the valuable information contained in reports, documents or natural language statements cannot be directly integrated into relational databases due to their rigid structure and predefined schemas. This limitation requires new approaches that can bridge the gap between unstructured text and structured data storage. One promising solution lies in the conceptual framework of Node of Knowledge (NOK), which provides a structured method for translating natural language knowledge into formats suitable for storage and retrieval in relational databases.

This paper presents a system for converting knowledge expressed in natural language into a relational database format. The transformation process is based on the NOK method, in particular its graphical representation known as Diagram Node of knowledge. For this purpose, a metamodel was developed that maps concepts from NOK diagrams to the concepts used in entity-relationship diagrams and thus enables a direct translation of DNOK structures into relational database entries.

A case study is used to show how knowledge from simple natural language sentences can be entered into a relational database.

Keywords. Knowledge representation, Node of Knowledge, NOK, DNOK, relational database

1 Introduction and related work

In business information systems, data is stored in relational databases. To date, various technologies have been developed to improve working with databases. There are many additional systems and tools of business intelligence, e.g. data warehousing tools (Subotic et al., 2014), (Ahmadi, 2023), which can improve the conclusion about data in the information system. However, there is a problem with storing information embedded in unstructured or semi-

structured texts (Pavlic, Jakupovic, et al., 2013). Only the data that is structured as prescribed by a database schema created in the design process according to a methodology of information system development is entered into databases. This means that only data that fulfils certain criteria and for which a number of restrictions apply (type of values, amounts, complex terms and similar) can be entered into a database. On the other hand, there is a great need for systems that enable the storage of textual knowledge (knowledge expressed in text form) in databases.

However, if we want to store the new knowledge in the form of natural language sentences, the question arises as to how natural language sentences can be entered into the relational database. The problem is how to enter knowledge from texts in books and natural language sentences into a relational database that allows us to perform semantic queries. If the sentence is entered into the text field, similar to entering text into editors, then it is not possible to perform the query over sentences, as is the case with the other data in the database. One of the methods that makes this possible is the Node of Knowledge (NOK) method, which is part of the Node of Knowledge conceptual framework

Large language models (LLMs) have recently dominated the field of natural language processing and question answering systems. Unlike knowledge bases, which involve large amounts of data, LLMs show a strong ability to generalise across a wide range of textual, tabular, general and even mathematical tasks to answer questions with few examples in context (Li et al., 2023). Currently, the best known and most widely used LLM is the GPT developed by OpenAI (ChatGPT, 2025). Large language models look for answers to queries that require a large amount of knowledge, which is why the accuracy of the answers obtained is questioned (Lewis et al., 2020), (Romano et al., 2024), (Athluri et al., 2023). Large language models use a large amount of unstructured knowledge to search for answers to questions, while the NOK conceptual framework uses structured knowledge. This makes the NOK method suitable for solving problems that require structured knowledge and a precisely selected set of knowledge in which answers to questions are sought.

The conceptual framework Node of Knowledge consists of a Node of Knowledge method and several formalisms (formalism for graphical representation – DNOK – diagram node of knowledge, formalized textual record – FNOK – formalized node of knowledge, formalized questions – qFNOK – question formalized node of knowledge).

The NOK method represents knowledge graphically in the form of diagrams of Nodes of Knowledge (Asenbrener Katic et al., 2015), (Rauker Koch et al., 2014), (Tomljanovic et al., 2014). The DNOK formalism (as well as other graphical methods of knowledge representation) shows knowledge in the form of a graph consisting of nodes and links. The nodes represent concepts and the links between the nodes are used to represent the relationships between the concepts. When comparing the NOK method with other graphical methods (Basic Conceptual Graphs (BG), Multi-layered extended semantic networks (MULTINET), Hierarchical Semantic Form (HSF), Resource Description Framework (RDF), it can be concluded that the NOK method is simpler than other methods (it has a smaller number of elements), more expressive (it allows the representation of knowledge at different levels of abstraction) and easier to read (in the NOK method it is possible to read the knowledge starting from any node, but using roles of relations) (Jakupovic et al., 2013). These characteristics of the NOK method enable easy deployment and thus broad applicability and are one of the main reasons why the NOK method was chosen.

The conceptual framework of NOK (Jakupovic et al., 2014) facilitates the transformation of knowledge from natural language texts into a Question Answering (QA) system by organising it into structured records suitable for integration into a relational database.

Studies (Rauker Koch et al., 2014), (Rauker Koch et al., 2015b), (Rauker Koch et al., 2022) have shown that the NOK conceptual framework enables the encoding of texts in multiple languages (Candrlic et al., 2020), into a formalised format suitable for input into a relational database. To achieve this transformation of natural language sentences into a structured format, it is necessary to analyse word types, define formalisation rules and develop a language metamodel. Previous research has looked at different word classes — nouns (Asenbrener Katic et al., 2021), verbs (Asenbrener Katic et al., 2018), adjectives (Pavlic et al., 2017) and adverbs and prepositions (Asenbrener Katic et al., 2022) and confirmed that the construction of a language metamodel and the encoding of sentences in a formalized record is indeed feasible.

Previous research has used the so-called one-way NOK method, in which the relationship between nodes is analysed in one direction only. In this paper, we present an extended variant of graphical knowledge representation that uses two-sided (or two-way) links, where each link carries two semantic roles. While DNOK also contains one-sided links — where a link has a single role — two-sided links offer greater

semantic richness and expressive power. Due to the characteristics of the two-sided NOK method, we believe that a question answering (QA) system based on it would provide better results.

This paper presents the two-sided NOK method, including the DNOK diagrams for the given examples and the corresponding data sets stored in a relational database.

After the introduction and presentation of related work in Section 1, the motivation for the research is presented in Section 2. Section 3 describes the research methodology. Sections 4 and 5 describe the NOK meta-model and the results of the case study. Section 6 provides conclusions and a plan for future work.

2 Research Motivation

The aim of the authors' research is to store knowledge derived from textual sources in a structured knowledge database that enables answers to questions to be generated on the basis of the stored information. The main motivation is to develop a QA system based on a relational database that can process and integrate knowledge from any text, regardless of the language in which it is written.

Previous research has used the so-called one-sided NOK method, in which the relationship between nodes is only analysed in one direction.

The question arises: How can two-role (two-sided) links be transformed into a relational database?

3 Methodology

The research methodology is based on the development of a metamodel for the two-sided NOK method and its representation as a relational database. Ten sentences were selected and transformed into DNOK diagrams. Based on the DNOK, tables were inserted into the database.

The Node of Knowledge method is based on researching and analysing natural human language. It analyses sentences, words and their meaning as well as the order in which words are linked together to form more complex thoughts. It is used to create knowledge models, i.e. to organise a knowledge network via ICT. NOK enables the storage of knowledge in a different way than language and writing, namely with the help of the human mind. The aim of the method is to enable the structuring of a knowledge network contained in a text form in any natural language (Pavlic, Mestrovic, et al., 2013). The basic concepts of the NOK method used in this are presented in Table 1.

Table 1. Basic concepts in the DNOK method

Concept	Symbol
Regular node	Iva
Process node	Drive
Link	Role 1 Role 2

Let us present the concepts of the NOK method by using a simple example. DNOK for sentence "A good Vedran drives a green car." is shown in Fig. 1.

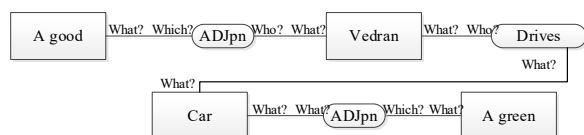


Figure 1. DNOK for sentence "A good Vedran drives a green car."

DNOK for this sentence consists of four regular nodes: "Vedran", "A good", "Car" and "A green" and the process node "Drives". Process nodes "ADJpn" are used on DNOK to connect two regular nodes whose link in a sentence is implied but is not explicitly stated. Nodes "Vedran" and "Car" are connected to the process node "Drives". In addition, the node "Vedran" is connected to the node "A good" with an adjective process node "ADJpn", while the node "Car" is connected to the node "A green" with another adjective process node "ADJpn". The nodes "A good" and "A green" in more detail describe the nodes "Vedran" and "Car" respectively, suggested by the process node "ADJpn". Two questions that define the role are listed on each link between nodes, and they are very important in storing and querying knowledge.

For example, the node "Vedran" and the process node "Drives" are connected by a link with two roles, "What?" and "Who?". If we ask "What Vedran?", The answer is "Drives". If we ask, "Who drives?" the answer is "Vedran". Node "Vedran" is also connected to an adjective process node "ADJpn" with a link with two roles, "What?" And "Who?". If we ask "What Vedran", the answer is "ADJpn". If we ask "Who ADJpn" the answer is "Vedran". The same process node "ADJpn" is on the other side connected by a link with two roles, "What?" and "Which?" with regular node "A good".

If we ask, "What a good?", the answer is "ADJpn". If we ask, "Which ADJpn?" the answer is "A good". In this case, the adjective process node, as previously mentioned, is used to connect two regular nodes, and from this we can conclude that node ADJpn, the adjective process node is used for a more detailed explanation of one of the regular nodes, i.e. node "Vedran" is determined in more detail by node "A good". The mechanism of the NOK method functioning is described in detail in previous papers on NOK method (Asenbrener Katic et al., 2015).

The next chapter presents a brief description of how simple sentences can be transformed into the DNOK model.

4 NOK metamodel

For the requirements of text transformation with the NOK method, it was necessary to define a model of the output database, the so-called NOK text database, into which the input data records are to be transformed and entered after application of the NOK method. For this purpose, a metamodel of the NOK method is defined, which is shown in Fig. 2. The NOK method enables recording natural language sentences in graphical form. By analysing the rules of the NOK method, the model of the NOK method (metamodel) can be represented using the Entity-Relationship method.

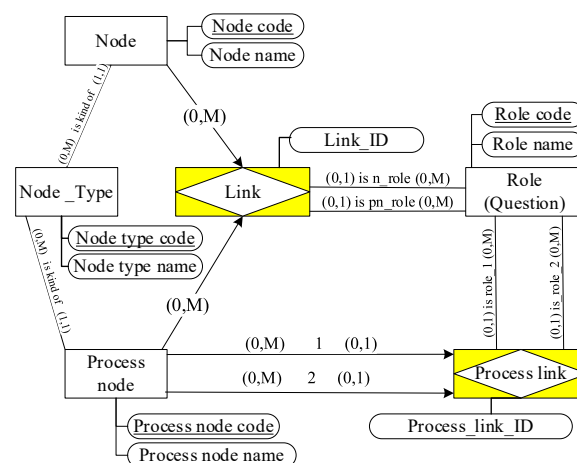


Figure 2. NOK Metamodel

Let us describe the proposed model in more detail. Entity type *Node_Type* contains a list of all possible types of nodes. A single occurrence of entity type *Node_Type* can have none or more occurrences of entity type *Node* and *Process node*. A single occurrence of the entity types *Node* and *Process node* belong to one and only one entity *Node_Type*. Entity types *Node* and *Process node* contain a list of all nodes in the NOK. This version of NOK method consists of two types of nodes, ordinary nodes (called simply nodes) and process nodes.

In NOK, there are links between nodes. A single occurrence of entity type *Node* can be connected to none or many occurrences of entity type *Process node*. A single occurrence of entity type *Process node* can be connected to none or many occurrences of entity type *Node*. Due to this, an aggregated entity type *Link* is introduced for the relationship between nodes, consisting of node and process node.

Each link has a relationship role to the node. The role (question) provides a clearer description of knowledge. These are usually interrogative pronouns and all

interrogative words that can be set. Each link has two roles (questions).

Entity type *Role (Question)* is connected with two relationships to the aggregated entity type *Link*. A single occurrence of entity type *Role* can be a role in none or more occurrences of aggregated entity type *Link*. A single occurrence of the aggregated entity type *Link* has two ends (for example, Node (Role_1) is connected to one, and Process node (Role_2) is connected to the other. The number of roles is (0,1), which means that a role may not have a name from the codebook.

In the case of the need to connect two entity types *Process node*, we come to the aggregated entity type *Process link*. Just like with the aggregated entity type *Link*, the aggregated entity type *Process link* is connected to the entity type *Role* with two relationships. A single occurrence of entity type *Role* can have none or more occurrences of aggregated entity type *Process link*. A single occurrence of aggregated entity type *Process link* belongs to none or only one entity *Role*.

Metamodel of the NOK method enables the transformation and recording of natural language sentences into a relational database using the NOK method.

Once we have entered the knowledge into the relational database, we have the possibility to store and query this knowledge using SQL queries. The metamodel provides enrichment of sentences with questions.

The relational database model (database schema) based on the metamodel NOK (from Fig. 2) is:

NODE_TYPE (Node type code, Node type name)

NODE (Node code, Node name, *Node type code*)

PROCESS NODE (PN code, PN name, *Node type code*)

ROLE (QUESTION) (Role code, Role name)

LINK (Node code, PN Code, Link_ID, *N_Role Code*, *PN_Role Code*)

PROCESS LINK (PN_1 Code, PN_2 Code, Process_link_ID, *Role Code_1*, *Role Code_2*)

5 Research Results and Discussion (Case Study)

This section will demonstrate how the input natural language sentences are entered into a relational database. According to rules for interpreting entity-relationship diagrams into a relational schema (Pavlić, 2011), based on the metamodel of the NOK method from Fig. 2, a relational database schema of the NOK method for storing text is defined. Algorithm for transformation of DNOK (with one-way link) is given in paper (Pavlic, Mestrovic, et al., 2013). Below you can see the transformation of two-way relationship of DNOK.

Examples of selected sentences in natural language that are transformed into DNOK diagrams are:

1. Peter reads a journal.
2. Mark drives a red car.
3. Mary's car is blue.
4. John drives a motor car.
5. A good Vedran drives a green car.
6. Jelena sits and reads a comic.
7. Tom swims.
8. The student reads a wonderful book.
9. Girls are on the beach.
10. Tom has two cars.

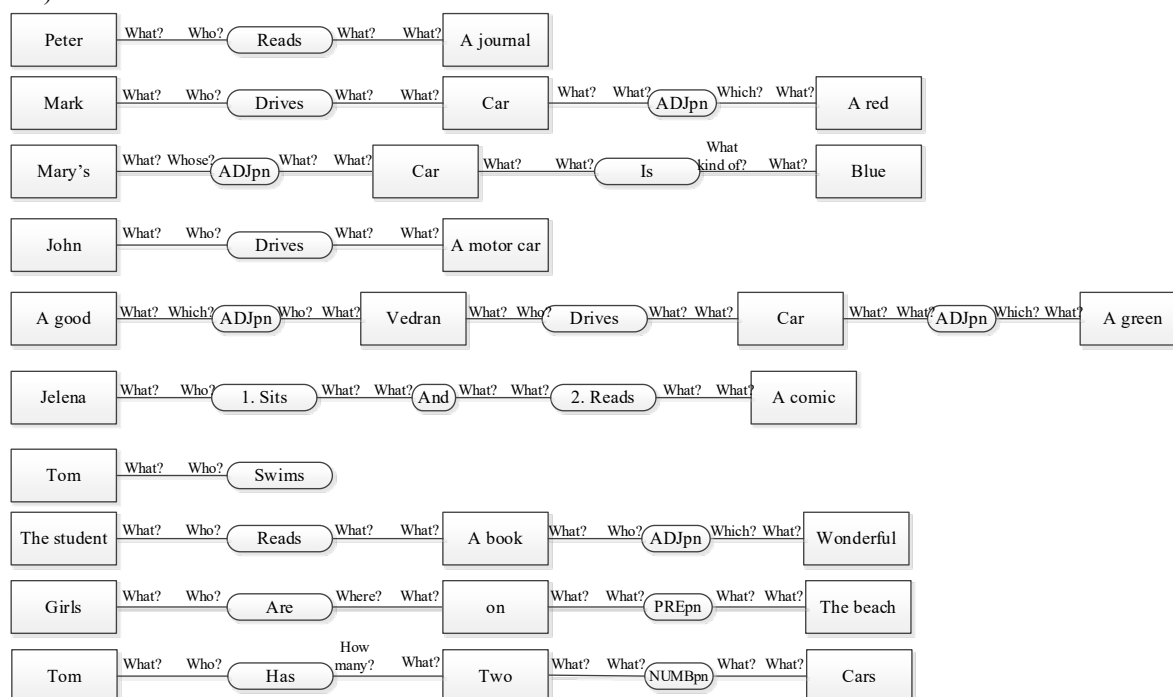


Figure 3. DNOKs for natural language sentences

In DNOK, articles "a", "an" and "the" are joined to words which they precede (while in the process of querying in SQL queries, those articles are ignored). Each row represents a single sentence.

The corresponding DNOKs for sentences in natural language are shown in Fig. 3.

The selected examples show different types of words and their modeling using the NOK method: nouns, verbs, adjectives, prepositions, conjunctions and numbers. Nouns and Verbs are two basic types of nodes. Adjectives, numbers and prepositions belong to nouns and are connected with special process nodes. Two types of adjectives are shown (definite and indefinite adjective) and their model in DNOK. Conjunctions are shown as process nodes.

Process nodes "ADJpn", "PREpn" and "NUMBpn", i.e. adjectival process node, prepositional process node and numeric process node respectively, in DNOK have no special semantics as words, but represent the class. They are inserted in order to connect two regular nodes, and also serve as a link between nodes that are closely related and enable the preservation of semantics in sentences. In the sentence: Jelena sits and reads a comic, "Sits" and "Reads" are the process nodes, with roles "what?" and "who?", as shown in Fig. 3. The conjunction "And" from this sentence is drawn on DNOK as a process node that connects the two process nodes.

Using the DNOKs from Fig. 3, we can fill in the tables based on the relational database schema (Tables 2-7).

Table 2. NODE TYPE

Node type code	Node type name
O	Regular node
P	Process node
K	Context node
ADJ	Adjectival process node
PRE	Prepositional process node
NUM	Numeric process node

Table 3. NODE TYPE

Node code	Node name	Node type code
a	Peter	O
c	A journal	O
d	Mark	O
f	A red	O
g	Car	O
h	Mary's	O
j	Car	O
k	Blue	O
l	John	O
n	A motor car	O
o	A good	O
p	Vedran	O
s	A green	O
t	Car	O
u	Jelena	O
x	A comic	O

aa	Tom	O
ac	The student	O
ae	A book	O
ag	Wonderful	O
ah	Girls	O
aj	On	O
ak	The beach	O
al	Tom	O
an	Two	O
ao	Cars	O

Table 4. PROCESS NODE

PN code	PN name	Node type code
b	Reads	P
e	Drives	P
e 1	ADJpn	ADJ
i	ADJpn	ADJ
i 1	Is	P
m	Drives	P
r	Drives	P
r 1	ADJpn	ADJ
r 2	ADJpn	ADJ
v	Sits	P
z	And	P
y	Reads	P
ab	Swims	P
ad	Reads	P
ad 1	ADJpn	ADJ
ai	Are	P
ai 1	PREpn	PRE
am	Has	P
am 1	NUMBpn	NUM

Table 5. ROLE (QUESTION)

Role code	Role name
I	Who
II	What
III	Whom
IV	What kind of
V	Where?
VI	How many?
VII	Which?
VIII	Whose?

The first five columns in Table 6 represent the content of relational database. The last two columns *Node Name* and *PN name* (*Process node name*), separated by an empty column, were introduced to facilitate monitoring of the transformation of the sentences' content record into the relational database. The same applies to the table 7. Process link.

Although a small number of sentences are involved, the data in Tables 2-7 show how knowledge from simple sentences in natural language can be entered into a relational database that allows knowledge to be queried and stored.

For example, if you ask "Who is driving?" based on the tables, we can get answers: "Marko", "Ivan" and "Vedran". If the question is expanded with another

node, we get more precise answers. For example, "Who drives a motor car?" The answer is "Ivan". To the question "Who is driving the car?" we will get answers, "Marko" and "Vedran", while the question "Who is driving a red car?" results in the answer: "Marko".

In addition, integration with the dictionary is required, allowing the inclusion of various semantic relations. In the case of the example shown, for example, one could ask: "Who drives automobile?". The information about automobile is not added directly to the database. However, according to the data from the monolingual dictionary, there is a connection between the word "car" and its synonym "automobile". Therefore, the answer would be: "Marko" and "Vedran".

The accuracy and concreteness of the answers (one or more) depends on how detailed the question is

formulated and how many nodes are used. Sometimes the answer is easy to find in the link and process link tables with one pass through the tables, while sometimes the answer requires several passes through the table, i.e. the complexity of the SQL query can vary.

6 Conclusion and future work

The article describes a system that enables the transformation of textually expressed knowledge into a relational database. The translation of natural language sentences into the relational database is based on the NOK method, i.e. the graphical formalism DNOK. In the paper, a metamodel was defined that enables the

Table 6. LINK

Link ID	Node code	PN Code	N Role Code	PN Role Code	Node name	PN Name
1.	a	b	II (What)	I. (Who)	Peter	Reads
2.	c	b	II (What)	II (What)	A Journal	Reads
3.	d	e	II (What)	I. (Who)	Mark	Drives
4.	g	e	II (What)	II (What)	Car	Drives
5.	g	e 1	II (What)	II (What)	Car	ADJpn
6.	f	e 1	II (What)	VII (Which)	A red	ADJpn
7.	h	i	II (What)	VIII. (Whose)	Mary's	ADJpn
8.	k	i	II (What)	II (What)	Car	ADJpn
9.	k	i 1	II (What)	II (What)	Car	Is
10.	j	i 1	II (What)	IV (What kind of)	Blue	Is
11.	l	m	II (What)	I. (Who)	John	Drives
12.	n	m	II (What)	II (What)	A motor car	Drives
13.	p	r 1	II (What)	I (Who)	Vedran	ADJpn
14.	o	r 1	II (What)	VII (Which)	A good	ADJpn
15.	p	r	II (What)	I. (Who)	Vedran	Drives
16.	t	r	II (What)	II (What)	Car	Drives
17.	t	r 2	II (What)	II (What)	Car	ADJpn
18.	s	r 2	II (What)	VII (Which)	A green	ADJpn
19.	u	v	II (What)	I. (Who)	Jelena	Sits
20.	x	y	II (What)	II (What)	A comic	Reads
21.	aa	ab	II (What)	I. (Who)	Tom	Swims
22.	ac	ad	II (What)	I. (Who)	The student	Reads
23.	ae	ad	II (What)	II (What)	A book	Reads
24.	ae	ad 1	II (What)	I. (Who)	A book	ADJpn
25.	ag	ad 1	II (What)	VII (Which)	Wonderful	ADJpn
26.	ah	ai	II (What)	I. (Who)	Girls	Are
27.	aj	ai	II (What)	V (Where)	On	Are
28.	aj	ai 1	II (What)	II (What)	On	PREpn
29.	ak	ai 1	II (What)	II (What)	The beach	PREpn
30.	al	am	II (What)	I. (Who)	Tom	Has
31.	an	am	II (What)	VI (How many?)	Two	Has
32.	an	am 1	II (What)	II (What)	Two	NUMBpn
33.	ao	am 1	II (What)	II (What)	Cars	NUMBpn

Table 7. PROCESS LINK

Process link ID	PN 1 Code	PN 2 Code	Role Code 1	Role Code 2	PN 1 Name	PN 2 Name
1	v	z	II (What)	II (What)	Sits	And
2	z	y	II (What)	II (What)	And	Reads

translation of concepts from NOK diagrams into concepts of entity-relationship diagrams. This has also enabled the direct transformation of data from DNOK into a relational database.

Finally, as a case study, we presented the input of knowledge into a relational database and explained it using the example of simple natural language sentences. In this paper, the NOK method is applied to simple sentences, in later stages we will apply it to complex sentences.

Querying an integrated information system with a new module (NOK system) or defining queries attached to both relational databases (database of the existing information system and database of the new NOK system) makes it possible to obtain answers to the complex question posed, which should be used to scan both text and data. For the query itself, various algorithms for entering data into the database must be defined, which requires great effort and further research.

The plan for further research includes analyzing more dictionary attributes and developing the dictionary model. Without the dictionary model, which uniquely names words, separates homonyms and semantically links synonyms, it is not possible to write algorithms required to develop the system for storing sentences in the relational databases. Our future goal is to develop a module for an information system based on the metamodel, which will enable the input of natural language sentences in two-way NOK records and their retrieval.

Acknowledgments

This work has been fully supported by the University of Rijeka under project uniri-iskusni-drustv-23-216.

References

- Ahmadi, S. (2023). Optimizing Data Warehousing Performance through Machine Learning Algorithms in the Cloud. *International Journal of Science and Research*, 12(12), 1859–1867. <https://doi.org/10.21275/SR231224074241>
- Asenbrener Katic, M., Candrljic, S., & Koch, M. R. (2022). Adverb and Preposition Representation in Croatian and English using the Node of Knowledge Method. *Central European Conference on Information and Intelligent Systems*, 41–48.
- Asenbrener Katic, M., Candrljic, S., & Pavlic, M. (2018). Modeling of verbs using the node of knowledge conceptual framework. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 1022–1027. <https://doi.org/10.23919/MIPRO.2018.8400187>
- Asenbrener Katic, M., Candrljic, S., & Pavlic, M. (2021). Nouns in the Conceptual Framework “Node of Knowledge.” *Tehnicki Vjesnik - Technical Gazette*, 28(6), 2088–2093. <https://doi.org/10.17559/TV-20200731095329>
- Asenbrener Katic, M., Pavlic, M., & Candrljic, S. (2015). The representation of database content and structure using the NOK method. *Procedia Engineering*, 100(January), 100(January). <https://doi.org/10.1016/j.proeng.2015.01.469>
- Athaluri, S. A., Manthena, S. V., Kesapragada, V. S. R. K. M., Yarlalagadda, V., Dave, T., & Duddumpudi, R. T. S. (2023). Exploring the Boundaries of Reality: Investigating the Phenomenon of Artificial Intelligence Hallucination in Scientific Writing Through ChatGPT References. *Cureus*, 15(4). <https://doi.org/10.7759/CUREUS.37432>,
- Candrljic, S., Asenbrener Katic, M., & Jakupovic, A. (2020). Preliminary multi-lingual evaluation of a question answering system based on the node of knowledge method. In *Lecture Notes in Networks and Systems* (Vol. 69). https://doi.org/10.1007/978-3-030-12388-8_69
- ChatGPT. (2025). *ChatGPT*. <https://chatgpt.com/>
- Jakupovic, A., Pavlic, M., & Dovedan Han, Z. (2014). Formalisation method for the text expressed knowledge. *Expert Systems with Applications*, 41(11), 5308–5322. <https://doi.org/10.1016/j.eswa.2014.03.006>
- Jakupovic, A., Pavlic, M., Mestrovic, A., & Jovanović, V. (2013). Comparison of the Nodes of Knowledge method with other graphical methods for knowledge representation. *Proceedings of the 36th International Convention /CIS/, MIPRO 2013*, 1276–1280.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems, 2020-December*. <https://arxiv.org/pdf/2005.11401>
- Li, T., Ma, X., Zhuang, A., Gu, Y., Su, Y., & Chen, W. (2023). Few-shot In-context Learning for Knowledge Base Question Answering. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 1*, 6966–6980. <https://github.com/ltl3A87/KB-BINDER>
- Pavlic, M., Dovedan Han, Z., Jakupovic, A., Asenbrener Katic, M., & Candrljic, S. (2017). Adjective representation with the method Nodes of Knowledge. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1221–1226. <https://doi.org/10.23919/MIPRO.2017.7973610>
- Pavlic, M., Jakupovic, A., & Meštrović, A. (2013). Nodes of knowledge method for knowledge representation. *Informatologia*, 46(3), 206–214.

- Pavlic, M., Mestrovic, A., & Jakupovic, A. (2013). Graph-based formalisms for knowledge representation. *Proceedings of the 17th World Multi-Conference on Systemic Cybernetics and Informatics (WMSCI 2013)*, 200–204.
- Rauker Koch, M., Čandrlić, S., & Ašenbrener Katić, M. (2022). Automation of the conversion of natural language to formalized node of knowledge record. *Zbornik Veleučilišta u Rijeci*, 10(1), 57–71. <https://doi.org/10.31784/zvr.10.1.4>
- Rauker Koch, M., Pavlic, M., & Asenbrener Katic, M. (2015). Homonyms and Synonyms in NOK Method. *Procedia Engineering*, 100, 1055–1061. <https://doi.org/10.1016/j.proeng.2015.01.466>
- Rauker Koch, M., Pavlic, M., & Jakupovic, A. (2014). Application of the NOK method in sentence modelling. *Proceedings of the 37th International Convention MIPRO 2014*, 1426–1431.
- Romano, S., Angius, R., Kerby, N., Bouchaud, P., Amidei, J., & Kaltenbrunner, A. (2024). A Dataset to Assess Microsoft Copilot Answers in the Context of Swiss, Bavarian and Hessian Elections. *Proceedings of the International AAAI Conference on Web and Social Media*, 18, 2040–2050. <https://doi.org/10.1609/ICWSM.V18I1.31446>
- Subotic, D., Jovanovic, V., & Posic, P. (2014). Data Warehouse Schema Evolution: State of the Art. *Central European Conference on Information and Intelligent Systems CECIIS*.
- Tomljanovic, J., Pavlic, M., & Asenbrener Katic, M. (2014). Intelligent question — Answering systems: Review of research. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1228–1233. <https://doi.org/10.1109/MIPRO.2014.6859755>