



# An Interactive Artificial Intelligence System for Inventive Problem-Solving

Stelian Brad<sup>1</sup>(✉) and Emil Ştetco<sup>2</sup>

<sup>1</sup> Technical University of Cluj-Napoca, Memorandumului 28, 400445 Cluj-Napoca, Romania

stelian.brad@staff.utcluj.ro

<sup>2</sup> Zetta Cloud, Govora 16A, 400535 Cluj-Napoca, Romania

emil.stetco@zettacloud.ro

**Abstract.** There is a vast space of potentiality for inspiration in the design and engineering of technical systems that are poorly valorized; the cyberspace that stores and daily adds high volumes of global collective intelligence. This space could be more productively tackled with the assistance of Artificial Intelligence algorithms led by Natural Language Processing (NLP) models. We investigate the application of Structured Activation Vertex Entropy (SAVE) method in combination with Question Answering Machine (QAM) algorithms to explore information that is stored in big datasets, accessible within unstructured dataspace. The SAVE method is transformed with the assistance of TRIZ into a set of searching meta-terms or meta-concepts. Taking off from a clear description of the problem, target results, and the current (eco)system, meta-terms, and concepts are incorporated into a spiral searching-answering process called ‘D-SIT-SIT-C’, driven by a Retrieval Augmented Generation (RAG) model to create an “intelligent” Natural Language Processing pipeline, with inserting the human in the loop at each iteration. We have found that the proposed pipeline based on a RAG model brings new valences to the creative thinking process and unleashes new dimensions of investigations that lead to higher quality solutions than those formulated with limited resources.

**Keywords:** Artificial Intelligence (AI) · Natural Language Processing (NLP) · Question Answering Machine (QAM) · Deep Learning (DL) · Theory of Inventive Problem Solving (TRIZ) · Structured Activation of Vertex Entropy (SAVE) · Retrieval Augmented Generation (RAG) · Humans in the Loop (HL) · Interactive AI systems

## 1 Introduction

With the development of algorithms for text mining that use regular expressions [1], as well as considering the improvements in natural language processing models (NLP) [2], researchers, scientists and engineers benefit a lot for searching in a smarter way within dataspace for specific information. This is more than using search engines and key words to display a list of sources, such as the case of Google or other similar web search

engines [3]. In inventive-problem solving frameworks we count on previous knowledge, discoveries, and inventions (e.g., patents, solutions published in scientific papers, etc.) to inspire us to solve a new problem. TRIZ explicitly encourages engineers to investigate knowledge bases at a given step in the problem-solving process [4], mostly databases with patents, where information is well-structured.

Case-based reasoning (CBR) is another approach to search for inspiration in problem-solving [5]. It is a model from the field of artificial intelligence (AI) and cognitive science that defines reasoning patterns to solve new problems by retrieving ‘cases’ that are stored in knowledge bases. In CBR, previous problem-solving experiences are extracted and adapted to fit new needs. CBR incorporates in the searching process a model of human reasoning as a mechanism to constructing intelligent systems [6].

Combinations of semantic models (which go beyond simple ontologies) and TRIZ have been considered to develop expert systems to search for ideas in structured patent databases and to make various analytics in the process of idea generation for a new problem. A representative technological development in this direction is PatentInspiration [7]. Semantic models with TRIZ inventive principles are embedded in another technology called InnovationQ Plus [8] that is designed to search in massive databases of patents and IEEE indexed scientific papers (over 100 million documents in all).

Matching and extraction of relevant knowledge from patent documents to be integrated within Inventive Design Method is another research contribution to support creative design [9]. In the same line, there are some other researches. Thus, our investigation in the mainstream databases with scientific publications (e.g., Clarivate Analytics, SCOPUS) revealed a research work about Artificial Intelligence (AI) driven inventions [10]. The authors cited in [10] articulate the idea that automating inventions look interesting, but they consider that AI will rather complement the intelligence of engineers and scientists, rather than replace it. They also present a semantic model that can form the basis of future AI approaches. As part of their work, over eight million patents and scientific publications have been screened with NLP techniques to extract semantic concepts. Another relevant research is about reproducing TRIZ reasoning through the Deep Learning (DL) on a large number of transdisciplinary patent sets [11]. The investigation of online data sources for pairing engineering problems with knowledge of physics for classification have also recently been published [12]. The last reference with relevance in relation to the topic considered in this paper is about the use of DL NLP-related models (Doc2vec and Cosine Similarity) for automatic information retrieval and introduction into an Inverse Problem Graph (IPG) process (note: IPG is a lean-based method for defining problems in the initial analysis phase of the inventive design process) [13].

The above-mentioned researches indicate a growing interest of the scientific community in introducing AI algorithms, mostly NLP DL-related ones, for smarter searching in databases on information and knowledge to support the inventive design process. However, our inquiry in mainstream databases of scientific papers does not indicate relevant research in the field of automating idea proposals for problem solving. This niche is going to be further elaborated on in the next sections of this paper. We consider the latest developments in artificial intelligence, specifically Question Answering Machine

(QAM) algorithms [14] in combination with the inventive principles promoted by Structured Activation of Vertex Entropy (SAVE) method [15] to assist the ideation process for engineering problems. This is about neuro-symbolic models of AI.

## 2 Methodology

Our aim is to investigate the application of Structured Activation Vertex Entropy (SAVE) method in combination with Question Answering Machine (QAM) algorithms to explore information that is stored in big knowledge/information/datasets (in a large number of documents of various formats), accessible within structured or unstructured dataspace (e.g., open Internet) and formulate an inventive solution to a problem.

The SAVE method is transformed with the assistance of TRIZ into a set of searching meta-terms or meta-concepts. Thus, by applying TRIZ contradiction matrix on the SAVE method in conjunction with QAM, the conflict is: how to transform SAVE for an easy operation (parameter 33 in TRIZ) without losing essential information (parameter 24 in TRIZ). This leads to the following set of TRIZ inventive principles (IP): asymmetry (IP 4); preliminary action (IP 10); cheap short-living objects (IP 27); turn harm into a benefit (IP 22). IP 27 inspired us to convert the ten vertexes of SAVE into key words, key actions, and target-related key words. IP 4 inspired us to include as many as possible synonyms to the key words. IP 10 encourages organizing the key words into some semantics. IP 22 was not considered in this research. With these indications, the results are as follows:

VERTEX 1: Activation of resonance [capable to resonate, work at the same frequency].

*Vertex 1 associated key words:* {resonance, resonant, reverberate, reverberation, harmony, harmonize, agreement, agreed, accord, consensus, unity, vibrate, vibrant, resonate, alignment, placement, configuration, positioning, pact, congruence, converge, convergence, synchronize, synchronization, pulsate, pulsating}. *Vertex 1 related key actions:* {capable, able} {work, operate, act, perform} + {resonant, same, similar, harmonic, natural} + {intervention, opinion, frequency, vibration}.

VERTEX 2: Introduction of neutral elements [capable to annihilate; activate a new path].

*Vertex 2 associated key words:* {annihilate, defeat, crush, overwhelm, overpower}; {path, trail, track, way, trajectory, direction, route, road, pathway}. *Vertex 2 related key actions:* {capable, able} + {annihilate, cancel, overwhelm, overpass, overcome}; {activate, work, create, generate, make} + {(new) + path, trajectory, direction, way, route, pathway}.

VERTEX 3: Action against the wolf-pack spirit [operate and reach a target with no support from other systems; operate and reach a target with a fully volunteer support from other systems].

*Vertex 3 associated key words:* {operate, reach, target, aim, objective, volunteer, support, system, structure, field, domain, no, without, other, external, single, alone, against}. *Vertex 3 related key actions:* {operate, activate, reach}; {target, goal, aim, desire, dream}; {(no, against, without) + support, assistance, help, guidance}; {alone, single}; {(full, entire, complete, big, large) + (volunteer, open, disinterest) + support, assistance, help, guidance}; {(other, outer, external) + system(s), domain(s), structure(s), field(s)}.

VERTEX 4: Activation of centrifugal forces [benefit from the dynamics of individual elements]. *Vertex 4 associated key words:* {benefit, advantage, activation, generation};

{speed, rotational, centrifugal, centripetal, dynamics}. *Vertex 4 related key actions*: {activate, generate, create, combine, produce} + {dynamics, element(s), force(s), field(s), motivation, (new) + (interest(s), attractor(s), influence(s))}.

VERTEX 5: Application of multi-level connections [act in alignment and synergy with other systems]. *Vertex 5 associated key words*: {action}; {synergy, alignment}; {multiple, external, other}; {system(s), module(s)}. *Vertex 5 related key actions*: {act, do, perform, execute, operate, align} + {synergy} + {system(s), unit(s), element(s), person(s), people, object(s), part(s), module(s)}.

VERTEX 6: Application of asymmetry [counterbalance a much bigger system]. *Vertex 6 associated key words*: {asymmetry, counterbalance}; {super-system, system}; {larger, bigger, wider, higher, longer, greater, comprehensive}. *Vertex 6 related key actions*: {equilibrate, leveraging, counterbalancing} + {activity, system, part, module, unit}.

VERTEX 7: Harmonization of individual goals with collective goals [aligned to a higher-level target]. *Vertex 7 associated key words*: {harmonization, alignment}; {performance, goal}; {improvement, radical}. *Vertex 7 related key actions*: {harmonize, aggregate, align} + {target, performance, goal} + {individual, collective}.

VERTEX 8: Transformation for value-added [can provide more outputs than before, using the same inputs]. *Vertex 8 associated key words*: {value-added, efficiency}; {input-output, transformation}. *Vertex 8 related key actions*: {transform, generate, create} + {efficiency, value-added}.

VERTEX 9: Application of prisoner paradox [use only the existing local resources; rearrangement, and utilization of local resources]. *Vertex 9 associated key words*: {reconfiguration, combination, rearrangement, reorganization}; {tool(s), material(s), element(s), resource(s)}; {local, limited}. *Vertex 9 related key actions*: {reconfigure, rearrange, combine, reorganize} + {local, limited, existent} + {resource(s), mean(s), tool(s), element(s), material(s), system(s)}.

VERTEX 10: Application of shipwrecked paradox [transform some local negative factors into positive factors, identify hidden resources]. *Vertex 10 associated key words*: {transformation, identification, disclosure, revealing}; {negative, positive}; {local, limited, existent}; {resource(s), factor(s)}. *Vertex 10 related key actions*: {dig, mining, discover, explore, investigate, search, identify, unhide, disclose, reveal, display, show}; {convert, transform, modify} + {negative, positive} + {factor(s), resource(s)}.

In addition, we consider the target-related key words representative for the particular use case. For the use case introduced in the Sect. 4 of this paper, the *particular target-related key words* are: {common, same, aligned} + {goal(s), interest(s)}; {no, low, missing, absent, irrelevant, little} + {obstacle(s), barrier(s), conflict(s), stress(es), tension(s), struggle(s), fight(s), impediment(s), problem(s), difficulty, complication(s), barricade(s), blockage(s)}.

To combine SAVE with QAM, we propose an algorithm that uses the problem, the system, and the target result in a series of transformative processes led by the SAVE vertexes and embedded in the QAM model. The algorithm is named D-SIT-SIT-C, meaning:

## DESCRIBE

1. Describe the problem, including synonyms for the keywords.

2. Describe the system, including synonyms for the keywords.
3. Describe the target result, including synonyms for the key target words.

## PROCESS FOR EACH VERTEX

### STEP 1: SEARCH

1. Search using the mix “system-problem”
2. Search using the mix “problem – target result”

### STEP 2: INVESTIGATE (WITH HUMAN-IN-THE-LOOP)

1. Investigate the extracted information – if it does not deliver acceptable solutions move to the next step

### STEP 3: TRANSFORM

1. Add to the system vertex-associated keywords
2. Add to the problem key actions

### STEP 4: SEARCH

1. Search using the mix “transformed system- transformed problem”
2. Search using the mix “transformed problem – target result”

### STEP 5: INVESTIGATE (WITH HUMAN-IN-THE-LOOP)

1. Investigate the extracted information – if it does not deliver acceptable solutions move to the next step

### STEP 6: TRANSFORM (WITH HUMAN-IN-THE-LOOP)

1. Create an intermediary solution by best possible use of the information collected in the previous steps
2. Make a new description of the TRANSFORMED system (intermediary solution)

### CONTINUE OR END

1. Take the next vertex and continue the process

As one can see, in the proposed algorithm human expert is present in the loop to analyze the results and to introduce additional inputs that would facilitate the progress

in problem resolution. Human-in-the-loop (HITL) is met in NLP, including QAM algorithms [16]. HITL in this paper is inspired from traditional online HITL, but it follows a different path. We do not consider HITL for improving the training dataset. We make corrections to the given answer and introduce the corrected answer in the QAM's loop without altering the training and testing datasets, and the trained model.

### 3 Results and Discussions

For our experiments we decided to use the recent advances in NLP domain with a pre-trained long form question answering system [17], which engages a question (engineering problem) and transformed with the assistance of SAVE method, fetches couple of relevant passages from the dataspace (e.g., in our case from Wikipedia snapshot), and writes a generated multi-sentence answer based on the question and retrieved passages. In particular, for our experiments we used the pre-trained ELI5 model from Hugging Face Model Repository [18], a model that was trained using the ELI5 dataset described in [19], and the Wiki Snippets Indexes [20] generated using the Wiki-40B: Multilingual Language Model Dataset [21] loaded from Hugging Face Datasets [22].

To implement the algorithm, we wrote the code in Python, using `eli5_utils.py`, `lfqa_utils.py` specific modules, as well as some general libraries such as `faiss_gpu`, `nlp`, `transformers`, and `torch`. The running environment was Jupyter Notebook, and for faster computation we used GPU resources (note: `import torch; torch.cuda.is_available()`).

As example, we selected a situation from the field of oil industry. Oil is deposited in huge reservoirs (e.g., 6 m height, 10 m radius or even bigger). Residuals settle over time to the bottom of these reservoirs and cover the heating pipes. To clean tanks from these residuals, oil must be extracted and deposited into new reservoirs. In this context, we can formulate an engineering challenge as follows:

*The problem:* how to clean the oil tanks from residuals collected on the tank's floor without extracting the oil from the tank, and without working manually to complete this kind of job.

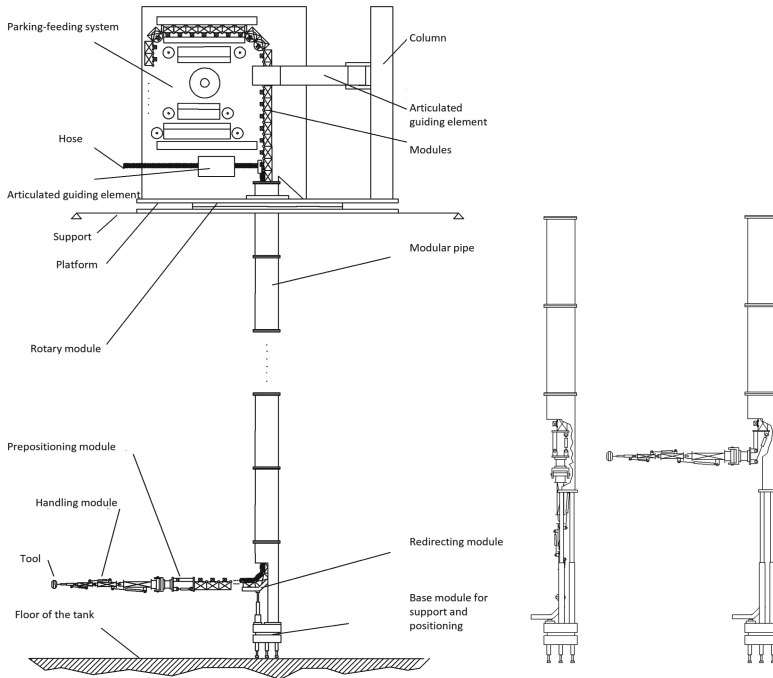
*The system at start:* oil tank, hole on the top of the tank, height of the tank, area of the floor, area of the hole, hole on the bottom side of the tank.

*The target result:* An automatic installation capable to enter a small hole on the top or bottom of the oil tank and cover a large surface on the floor of the tank to extract residuals.

First, we will use ARIZ [23] to find a solution to this problem. After that we will apply the proposed AI-driven algorithm to see the results. Discussions around results with the two approaches (a traditional one and one driven by AI for searching within the space of collective intelligence) will be also considered.

ARIZ recommends separating the opposite properties over time. The characteristics to be considered are: "12. Shape" (seen as the robot configuration in this case), which conflicts with: "8. The volume of the static object". The following inventive principles result: "7. Nest-in-nest: an object is placed inside another object, etc.; through a cavity, an object passes into another object"; "2. Extraction: extracts, removes, or separates a

part or property from the object that is bothering it”; “35. Transforming object properties: changing the degree of flexibility”. Results are shown in Fig. 1.



**Fig. 1.** Solution generated with indications from ARIZ application.

Now, we consider the D-SIT-SIT-C algorithm for the same problem. Intentionally, we will not consider all resources in system description to see how the D-SIT-SIT-C algorithm is dealing with. This means, we will intentionally exclude from system description the hole from the bottom side of the tank just to be somehow aligned with the solution proposed in Fig. 1, with the help of ARIZ method. We expect that the AI model will search in the database for other resources, including those excluded by us.

## DESCRIBE

*Describe the problem*, including synonyms for the *key words*: how to [clean, suction, extraction] the [oil, petroleum, gasoline] [tanks, reservoir] from [residuals, waste] [collected, deposited] on the [floor, bottom] without [extracting, emptying] the oil from the tank, and without working manually.

*Describe the system*, including synonyms for the *key words*: height of the tank, area of the floor, area of the hole, robot structure, robot joints, robot arm.

*Describe the target result*, including synonyms for the *target key words*: A [robot, technology, mechanism, device] capable to enter a [small, tiny] [hole, hollow, cavity] on the [top] of an oil tank and cover a large surface on the floor of the tank.

## STEP 1: SEARCH

*Search* using the mix “system-problem”. *Question*: the text from “Describe the problem” + the text from “Describe the system”.

The Wiki top 10 support documents retrieved by the system are not included here because of space considerations, but we can reveal the areas indicated in these documents: mud agitator, desander solids control, triplex-style plunger pumps & compressed-air-powered double-diaphragm pumps, industrial extraction with centrifugal systems, artificial lift with rod less pumping & continuous belt transportation, electrostatic coalescer, automated tank cleaning machine from chemical industry, Petrosix oil extraction process, artificial lift with hybrid gas and rod pump, artificial lift with electric submersible pumps (ESP).

[eli5 + Wiki] *automatic answer 1*: “You don’t need to suction the oil out of the tank, you just need to remove the oil from the tank. You can do this by using a pump to suck up the oil, and then you can pump the oil back into the tank with a pump attachment. The pump attachment is a bit more complicated, but the basic idea is the same.”

We want to highlight the clear, human-like answer automatically formulated by the AI-driven model. This is in our opinion a step forward in helping the creative problem solving of untrained people in structured innovation.

*Search* using the mix “problem – target result”. *Question*: the text from “Describe the problem” + the text from “Describe the target result”.

The Wiki top 10 support documents reveal two new indications: multi-stage gas extractor, and enhanced stripping methods from glycol dehydration.

[eli5 + Wiki] *automatic answer 2*: “There are a lot of different ways to do it, but the most common is to pump the oil out of the tank or reservoir, and then pump it back into the tank/reservoir. You can also pump it into the reservoir, but that’s more expensive than just pumping it out. There are also ways to remove the oil from the reservoir and pump it out, but those are more expensive.”

## STEP 2: INVESTIGATE (WITH HUMAN EXPERTS)

*Investigate* the extracted information – if it does not deliver acceptable solutions move to the next step. In our case we decided to move to the next step just to explore the space of potentiality, even if relevant information has been revealed at STEP 1.

## STEP 3: TRANSFORM

*Add* to the system *SAVE vertex-associated key words* (in this step VERTEX 1). *Question*: [eli5 + Wiki] *automatic answer 1* + [resonance, harmony, vibrate, configuration, positioning, converge].

The Wiki top 10 support documents retrieved in addition to the previous information other new directions of investigation: fluid coupling with scoop control, multi-stage centrifugal pumps & centrifugal pumps for solids control & magnetically coupled pumps, hit-and-miss engine, velocity triangle & vertical centrifugal pumps & froth pumps, hot-bulb engine, fluidyne engine.

[eli5 + Wiki] *automatic answer 3*: The answer retrieved is identical with answer 1. It looks that the system is not capable to provide additional insights.

*Add key actions* in conjunction to problem. *Question*: the text from “Describe the problem” + [activate, generate, create, combine, produce] + [dynamics, element(s), force(s), field(s), (new) + (interest(s), attractor(s), influence(s))].



The Wiki top 10 support documents retrieved by the system indicates other new directions of investigation: Lurgi–Ruhrgas process, RDF-power station, petroleum microbiology, Carabobo field.

[eli5 + Wiki] *automatic answer 4*: “You don’t need to do any of those things. You just need to be able to get the oil out of the tank or reservoir. It’s not like you have to do anything to get it out, *it’s just a matter of getting it out fast enough* that it doesn’t build up to the point where it’s a problem.”

With the indications from the last answer, we made some transformations to the level of the answer and the problem. This process is required by the SAVE algorithm, in order to move forward with the investigation.

*Transformed answer*: It’s just a matter of getting the residuals fast enough that it doesn’t build up to the point where it’s a problem.

*Transformed problem*: Is there a robot that cleans an oil tank from residuals without extracting the oil?

#### STEP 4: SEARCH

*Search* using the mix “transformed system- transformed problem”. *Question*: “Is there a robot that cleans an oil tank from residuals without extracting the oil? It’s just a matter of getting the residuals fast enough that it doesn’t build up to the point where it’s a problem.”

The Wiki top 10 support documents retrieved by the system indicates other new directions of investigation: hydropneumatic suspension, bioclogging, green nanotechnology for cleaning up oil spills, dry milling and fractionation of grain, oily water separator (marine), Alberta Taciuk process.

It looks like the deeper we investigate the problem indications move towards chemical processes rather than mechanical systems. This is a paradigm shift from the initial way of setting up the target result. This means, the AI system was not bordered by the limited perception (psychological inertia) of the team. It is also interesting the answer generated by the system.

[eli5 + Wiki] *automatic answer 5*: “Yes, there is (about the robot). *It’s called a scrubber*, and it’s used to remove the residuals from the oil tank. The problem is that scrubbers don’t remove all of the oil. Some of it is still in the tank, and the scrubber can’t get rid of all of it, so it builds up to the point where it’s a problem.”

Thus, the system was able to “judge” the relevance of having a pure mechanical solution and to “decide” that it is not necessarily the optimal one.

*Search* using the mix “transformed problem – target result”. *Question*: “It’s just a matter of getting the residuals fast enough that it doesn’t build up to the point where it’s a problem [robot, technology, mechanism, device] capable to enter a [small, tiny] [hole, hollow] on the top of an oil tank and cover a large surface on the floor of the tank”.

The Wiki top 10 support documents retrieved by the system indicates other new directions of investigation: Nokomis 3 process, rebreather with CO<sub>2</sub> scrubber, artificial pressure, annular fluidized bed with fine particulates, Finnish pollution control vessel Louhi.

[eli5 + Wiki] *automatic answer 6*: “It’s not just a matter of getting the residuals fast enough that it doesn’t build up to the point where it’s a problem. The problem is that there’s no way to get rid of it fast enough for it to be a problem, so it builds up

until it's too big to be contained." The answer shows that the system did not advance in "judgment". This somehow indicates that the searched resources have been covered.

#### STEP 5: INVESTIGATE (WITH HUMAN-IN-THE-LOOP)

*Investigate* the extracted information – if it does not deliver acceptable solutions move to the next step. In principle, for the problem exemplified here, the system was able to reveal sufficient advices and directions to think to a novel solution. However, we can investigate further, sometimes just for the sick of curiosity and consolidation of the decision.

#### STEP 6: TRANSFORM (WITH HUMAN-IN-THE-LOOP)

*Create* an intermediary solution by best possible use of the information collected in the previous steps. In this case, the intermediary formulated solution is: "Pump the oil out of the bottom of the tank, filter it and then pump it back as quickly as possible into the tank from the top of it. The installation that cleans an oil tank from residuals without extracting the oil is called the scrubber [24]."

*Make* a new description of the *transformed system* (intermediary solution). The new system includes scrubber, pump, handling unit, chemical reactions, filters, units that generate fields or forces.

#### CONTINUE OR END

Take the next SAVE vertex and *continue* the process. In the second cycle we have considered VERTEX 2. The detailed steps are not further described in the paper. In continuation we limit ourselves to present only the last relevant answer generated by the system.

[eli5 + Wiki] *automatic answer 7*: "Pumping the oil out of the bottom of the tank, filtering it, and then pumping it back as quickly as possible into the tank from the top of it. The robot that cleans an oil tank is called the Crude Oil Washing (COW). The COW washing system, or COW system, is part of the cleaning process."

Now it's the time to use the list of information and indication to ideate the solution to the problem. Actually, the last answer is a solution to the problem, automatically indicated by the system. We have also considered additional indications from the list retrieved by the system. It cannot be disclosed here because of an ongoing research project with inventive mission.

Comparing the solution generated with the help of ARIZ method and the one proposed by the system we can conclude that the first one is novel in the series of solutions destined to petroleum tank cleaning, whereas the second is extracted from the pool of knowledge already created by humans. However, the first solution was generated with a huge creative contribution of human experts, and it is a pure mechanical one. In the case of the AI-driven approach we did not want to develop a new tool for ideation, but rather a tool that helps engineers in the application of ARIZ or similar algorithms for inventive problem. It is a more powerful form of CBR because it moves beyond just displaying existent solutions, meaning it suggests vectors of investigation, as actually the TRIZ does, but with much more practical indications. These vectors of investigation are extracted from the documents retrieved in various steps of the proposed algorithm, as one already saw in the previous paragraphs of this section. They can be combined using morphological charts to help engineers thinking to various options for ideation.

## 4 Conclusions

This research introduces a possibility to transform a human-like nonlinear pattern of deep thinking into a format that is suitable for implementation into AI NLP-driven algorithms that search throughout big information/data/knowledge sets and automatically generate “intelligent” machine answers to a given problem. SAVE method is embedded in the algorithm to increase the search effectiveness and provoke the system in answer formulation. But there are no limitations to replace SAVE vertexes with TRIZ inventive principles or to add them to the vertexes list. The single issue is the productivity of doing the job.

The use case illustrated in the paper reveals a sign of encouragement, indicating that AI models of QAM can be a reliable assistant for experts in solving complex technical problems in shorter periods of time and at higher levels of performance. In fact, this research brings a new tool to the world of ARIZ, where the use of knowledge bases is required for documentation. Traditional software systems were limited in providing a massive pool of resources. Modern tools such as those cited in [7] and [8] better fulfill this job. In the case of our model, we claim that it brings something more, by also indicating basic solutions or clues towards the solution in a very natural language. Our research also indicates that there are still huge steps until the automatic systems will be able to replace human mind and become sufficient creative. We see the current systems rather being more explorative and highly productive in doing this job than people, and less creative in the sense of proposing out-of-the-box ideas. But they can help in indicating disruptive vectors of investigation, which is a relevant finding of this research. This means, we should not be limited in the short list of 40 generic inventive principles proposed by traditional TRIZ, and expand its pool of suggestions, moving towards the modern TRIZ.

It is important to highlight the fact that the capacity of the AI system is limited by the pool of training data, and from this angle one area to support inventive problem solving is to work on a dataspace of information that collects as many as possible information created by humankind (e.g., patents, open Internet, scientific papers, images, videos). Thus, new spaces for AI are open in this regard. Further effort is needed to expand these experiments on various knowledge bases and online resources, including scientific web databases or the unstructured open Internet, with making sure that not only the English language content is targeted but also other languages, as well. For this we would need to train our own dense models from within the knowledge base and also sequence-to-sequence models which are built upon a BART architecture [25] domain and language specific. Our future researches in the direction opened by this paper are also to train new models capable to deal with managerial problems, as well as to refine the algorithm in terms of formulating questions, etc. We also see a space of opportunity in testing some other tools of creativity in conjunction to QAM. Of course, the challenge to create a smarter system, capable to play with creative patterns is also in view for our future researches.

## References

1. Friedl, J.: *Mastering Regular Expressions*, 3rd edn. O'Reilly, Sebastopol (2006)

2. Lane, H., Hapke, H., Howard, C.: *Natural Language Processing in Action*. Manning, New York (2019)
3. Dwyer, D.: Top 12 Best Search Engines in the World (2016). <https://www.inspire.scot/blog/2016/11/11/top-12-best-search-engines-in-the-world238>. Accessed 20 June 2022
4. Gadd, K.: *TRIZ for Engineers*. Wiley, Chichester (2011)
5. Schmidt, R., Montani, S., Bellazzi, R., Portinale, L., Gierl, L.: Cased-based reasoning for medical knowledge-based systems. *Int. J. Med. Inform.* **64**, 355–367 (2001)
6. Lee, C.H., Chen, C.H., Li, F., Shie, A.J.: Customized and knowledge-centric service design model integrating case-based reasoning and TRIZ. *Expert Syst. Appl.* **143**, 13062, 14 pp. (2020)
7. Dewulf, S., Childs, P.R.N.: Patent data driven innovation logic: textual pattern exploration to identify innovation logic data. In: Borgianni, Y., Brad, S., Cavallucci, D., Livotov, P. (eds.) TFC 2021. *IAICT*, vol. 635, pp. 170–181. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86614-3\\_14](https://doi.org/10.1007/978-3-030-86614-3_14)
8. Ip.com: Why Non-Patent Literature Can Make or Break Your Business. [https://ip.com/wp-content/uploads/2020/09/IQ\\_NPL\\_ebook\\_P2.pdf](https://ip.com/wp-content/uploads/2020/09/IQ_NPL_ebook_P2.pdf). Accessed 02 June 2022
9. Souilia, A., Cavallucci, D., Rousselot, F.: Natural Language Processing (NLP) - a solution for knowledge extraction from patent unstructured data. *Proc. Eng.* **131**, 635–643 (2015)
10. Kaliteevskii, V., Deder, A., Peric, N., Chechurin, L.: Concept extraction based on semantic models using big amount of patents and scientific publications data. In: Borgianni, Y., Brad, S., Cavallucci, D., Livotov, P. (eds.) TFC 2021. *IAICT*, vol. 635, pp. 141–149. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86614-3\\_11](https://doi.org/10.1007/978-3-030-86614-3_11)
11. Guarino, G., Samet, A., Cavallucci, D.: Patent specialization for deep learning information retrieval algorithms. In: Borgianni, Y., Brad, S., Cavallucci, D., Livotov, P. (eds.) TFC 2021. *IAICT*, vol. 635, pp. 162–169. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86614-3\\_13](https://doi.org/10.1007/978-3-030-86614-3_13)
12. Boufeloussen, O., Cavallucci, D.: Bringing together engineering problems and basic science knowledge, one step closer to systematic invention. In: Borgianni, Y., Brad, S., Cavallucci, D., Livotov, P. (eds.) TFC 2021. *IAICT*, vol. 635, pp. 340–351. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86614-3\\_27](https://doi.org/10.1007/978-3-030-86614-3_27)
13. Hanifi, M., Chibane, H., Houssin, R., Cavallucci, D.: Problem formulation in inventive design using Doc2vec and cosine similarity as artificial intelligence methods and scientific papers. *Eng. Appl. Artif. Intell.* **109**, 104661 (2022)
14. Hugging Face: What is Question Answering?. <https://huggingface.co/tasks/question-answering>. Accessed 04 June 2022
15. Brad, S.: Domain analysis with TRIZ to define an effective “Design for Excellence” framework. In: Borgianni, Y., Brad, S., Cavallucci, D., Livotov, P. (eds.) TFC 2021. *IAICT*, vol. 635, pp. 426–444. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86614-3\\_34](https://doi.org/10.1007/978-3-030-86614-3_34)
16. Wang, Z.J., Choi, D., Xu, S., Yang, D.: Putting humans in the natural language processing loop: a survey. <https://arxiv.org/abs/2103.04044> (2021). Accessed 04 May 2022
17. Roy, A.: Progress and Challenges in Long-Form Open-Domain Question Answering. <https://ai.googleblog.com/2021/03/progress-and-challenges-in-long-form.html>. Accessed 03 Apr 2022
18. Jernite, Y.: ELI5 Model from Hugging Face Model Repository. <https://huggingface.co/yjernerite>. Accessed 02 Feb 2022
19. Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., Auli, M.: ELI5: Long Form Question Answering. <https://arxiv.org/abs/1907.09190> (2019). Accessed 20 Jan 2022
20. Wikipedia: User scripts/Snippets. [https://en.wikipedia.org/wiki/Wikipedia:User\\_scripts/Snippets](https://en.wikipedia.org/wiki/Wikipedia:User_scripts/Snippets). Accessed 05 Apr 2022

21. Guo, M., Dai, Z., Vrandečić, D., Al-Rfou, R.: Wiki-40B: multilingual language model dataset. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 2440–2452. European Language Resources Association, Marseille, France (2020)
22. Hugging Face Data Sets. <https://github.com/huggingface/datasets>. Accessed 05 Apr 2022
23. Cameron, G.: ARIZ Explored: A Step-by-Step Guide to ARIZ, the Algorithm for Solving Inventive Problems. Create Space, Scotts Valley (2015)
24. Wikipedia. Scrubber: <https://en.wikipedia.org/wiki/Scrubber>. Accessed 02 June 2022
25. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. <https://arxiv.org/abs/1910.13461> (2019). Accessed 04 May 2022