Towards Human-level AI Through Self-supervised Knowledge Learning and Semantic AI Microservices Engineering

Hong Qing Yu[1], Sam O'Neill[1]

School of Computing and Engineering

University of Derby, UK[a,1]

[a]*Markeaton Street, Derby, DE22 3AW*

Abstract

To create a human-level AI system, you must answer the first question 'Can machines think?'. This was a question asked by Turing in 1950 that started modern AI research. The fast development of machine learning technology (especially the Deep Learning paradise) shows that machines can provide the solution to a specific problem by learning data statistic features, a kind of intelligence similar to pattern recognition. However, the question presented by Turing remains unresolved and is still one of significant debate. The core reason is that human-level thinking is vastly more complicated than a fixed model tailored to a specific problem. Therefore, many giant AI research groups began to propose knowledge-focused next-generation AI systems. In this research paper, a self-supervised knowledge learning framework with semantic AI microservices engineering is proposed to build an AI system that can contextually solve problems in a manner more akin to human thinking. The paper details research gaps and core components, complete with an implementation design to address them. There are three layers - knowledge request, knowledge reasoning, and knowledge graph. The thinking method adapts the reinforcement learning approach to create a knowledge policy to guide system actions for different AI

Email address: h.yu@derby.ac.uk, s.oneill@derby.ac.uk (University of Derby, UK)
URL: www.derby.ac.uk (University of Derby, UK)

tasks. We developed a proof-of-concept prototype that follows the proposed vision. The prototype demonstrates that the proposed framework is one of the possible paths toward developing a more human-level AI system by evaluating different types of real-world application scenarios.

Keywords: Artificial Intelligent, Knowledge-based System, Microservice Engineering, Multiple-tasks AI, System Automation

1. Introduction

'Can machines think?' asked by Turing [1] opens up a significant question to researchers about whether a human-level AI system can be achieved. The research attracts many generations of researchers across numerous scientific disciplines, such as Computer Science, Life Science and Engineering. With the broad usage of Deep Neural Networks (DNN) applied to Big Data, researchers have started to ponder whether we are closer to achieving human-level AI systems. However, these DNN applications still show weak AI characteristics that only work on one specific task with a defined dataset. The major problem is that the current AI systems (algorithms or models) do not have the self-learning and reasoning capability to guide the behaviours of a system. It is important to define different levels of AI clearly, before presenting the research hypotheses. We adopt the following based on the descriptions are given in [2] to define three levels of AI characteristics:

- **Level 1: Data statistics oriented instinct AI** - Level 1 has a slow evolution process to update the model through weight-fixing and can only deal with one specific task. Most current AI systems like DNN can be categorized into this level.
- **Level 2: Continued learning and context-aware reasoning AI** - Level 2 has an efficient self-learning mechanism to carry on knowledge (e.g. behaviour policy or concept understanding) acquisition and storage. In the end, the system can solve similar problems to deal with different tasks

using existing knowledge to adopt new behaviours based on the context. Some robot systems and self-driving car research outcomes can partially satisfy these characteristics.

- **Level 3: Strong AI** - The extra communication (language) to learn knowledge from machine experiences and perform creative behaviours. There are a lot of critical debates at this level related to morals, necessity, and threats to human security. These topics are crucial but not the main focus of this paper.

Our research aims to do exploration on a knowledge-led semantic AI Microservices (AIMSs) environment that develops level 2 and partial level 3 AI systems. We have two major assumptions (pre-conditions):

- AI-related algorithms, models and processing functions can be decoupled and treated as microservices. A microservice is a service that has a unique and one-purpose function, e.g. loading data, importing an image, or getting a machine learning model. In addition, the service has a limited number of input parameters (maximum of three) and just one output. However, the output can be a complex object. In addition, AIMSs can be deployed in both clouds and local computing space with an invokable URI.

- There is a well-defined controlled ontology that describes AIMSs and its knowledge. We will define the top layer ontology in this paper and all the vocabulary can be retrieved from DBpedia Linked Data Cloud [3].

This research hypothesizes that the machine can continuously perform self-knowledge learning while AIMSs auto-configures the solution to a given task. Later, knowledge acquired from these experiences will drive optimisations of the solution.

The rest of the paper contains five sections. Section 2 presents our vision and current related work explain focusing on Google, Meta, and AutoKeras's visions

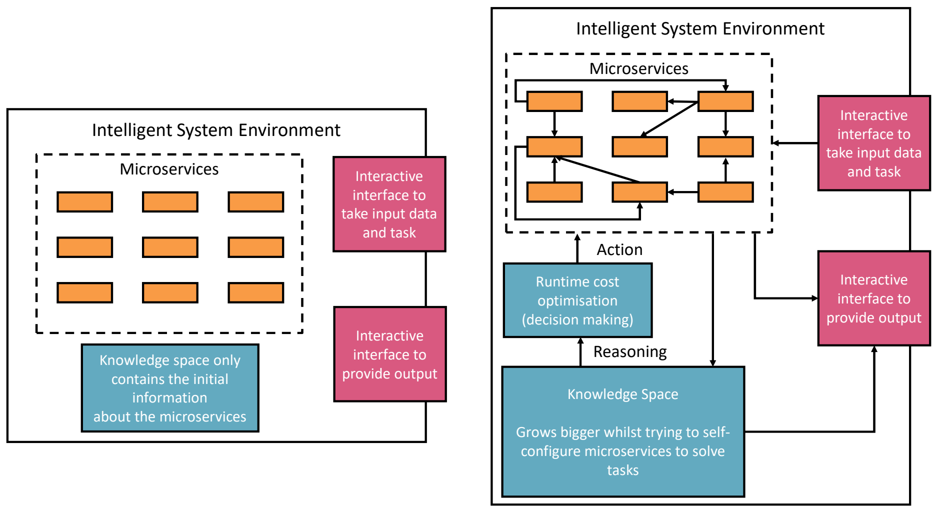

Figure 1: The vision of the self-knowledge learning approach with AI Microservices

and their early experiments. Section 3 introduces self-supervised knowledge learning modelling and design. Section 4 details the implementation of the experimental prototype. Section 5 evaluates the prototype works on four real-world AI tasks and discusses the lesson learned. The last section concludes our research work.

2. Our Vision and Related Work

2.1. The vision

Figure 1 represents our vision of Self-supervised Knowledge Learning with the AIMSs Engineering approach. The left part of the figure presents the initial settings of the intelligent environment. The initial environment only contains default AIMSs information such as purposes, I/O requirements, and invokable URI (detailed AIMS metadata ontology will be introduced in the next section). However, the initial settings are ready for doing four things:

- Registering new AIMSs from outside the environment. The registration process is through the interactive interface according to the defined microservice ontology. Therefore, human involvement in AI microservice engineering is a core part of this vision, which defines humans as an educator to teach basic skills and capabilities to deal with different tasks. Then, the environment will reuse these skills and capabilities to acquire knowledge. The knowledge will provide powerful reasoning sources to independently deal with complex tasks, decision-making, and creating new pipelines.

- Taking tasks with a variety of inputs, such as CSV data files, images, text, audio data, and videos. The environment auto-configures on the default AIMSs and provides solutions to the tasks. The success or failure outcome will be recorded as knowledge. The microservice human engineering process will start if there are no suitable AIMSs to deal with the task.

- The environment can compose multiple AIMSs to complete a task if one microservice cannot achieve it.

- The environment can start learning, representing, and storing knowledge in the knowledge space as knowledge graph data. The knowledge is derived from processing input data, the auto-configuration process, and task outcomes. The knowledge size will increase and thus provide better optimisations, auto-configuration, and feedback to the system user.

To realise the vision presented in Figure 1, we will discuss the related existing technologies and their research outcomes that can be adapted into our research next.

2.2. Multiple-task AI system research

Industrial AI leading research groups such as Google AI and Meta AI understood that data-driven AI technologies have issues with performing complex tasks. For example, creating human conversations with contextual understanding, or detecting early signs of disease from images. In addition, data-driven

AI is resource intensive and suffers from algorithm bias [4]. Thus, the multiple-task enabled AI systems with a knowledge-driven approach present a pathway toward a solution to these problems. Why is it thought that a knowledge-driven approach is necessary and crucial for the multiple-task system? There are two reasons:

- The information acquired from different tasks may present value that can be used as the basis to build new ML models for new tasks without requiring the high-cost processing to re-capture the same feature characteristics.
- Updating knowledge through validation is a relatively consistent process that will be less prone to bias from noisy data.

On completion of this research, two new ideas from Google and Meta have been published.

Google present an experimental process based on knowledge-mutation [5, 6]. Here the knowledge refers to base neural network transformers. To begin with, the experimental environment contains transformers which can work on different tasks (different image datasets for the classification problems). Then, when a new task arrives, the most related transformer will be triggered to do a mutation process. The mutation process can edit the base model by inserting a new layer, removing a layer, or doing both according to the performance optimisation. In the end, a new mutated adapter is created to enable dealing with a similar task next time. Whenever a new task with a new dataset arrives, the mutation process is executed based on the latest mutated model.

Meta research group presents a world model approach to acquiring knowledge very much in the spirit of actor-critic reinforcement learning [7]. The system architecture is a combination of smaller modules - configurator, perception, world model, cost, short-term memory, and actor - that feed into each other. The world model module is responsible for maintaining a model of the world that can then be used to both; estimate missing information about the world, and predict plausible future states of the world. The perception module will receive signals to estimate the current state of the world and for a given task the configurator module will have trained the perception module to extrapolate the relevant signal information. Then in combination, the perception, world model, cost, short-term memory, and actor modules feed into the configurator module which configures the other modules to fulfil the goals of the task. Finally, the actor module is handed the optimal action to perform as an action. This has an effect on the real world which the perception module can then capture which in turn triggers the process to repeat. That is, each action will produce a piece of state-changing knowledge feedback to the world model for continuous learning.

Both Google and Meta's visions derive from the previous hyperparameter optimisation-based AutoML processes [8], For example, AutoKeras [9], a neural architecture auto-search framework is proposed to perform network morphism guided by Bayesian optimisation and utilising a tree-structured acquisition function optimisation algorithm. The searching framework selects the most promising Keras implemented NN for a given dataset.

The above experimental results show improvements in tackling complex AI tasks and possible pathways toward human-level AI systems. However, there are two main limitations:

- The knowledge definition is too narrow and only uses the generated neural network as the knowledge limits the capability of recording all valuable outcomes through the learning experience.
- There is no unified knowledge representation structure for knowledge inference (machine thinking).

Do we already have a knowledge representation framework from our AI research in the past 70 years? The answer is yes.

2.3. Knowledge representation and reasoning

Knowledge representation and reasoning (KRR) are always the core research areas in AI systems [10]. The knowledge-enhanced machine learning approach attracts less attention that the data-driven approaches. However, KRR is still key in developing the future generation of AI systems development [11], even

Deep Neural Networks (DNN) can create KRR but just in a different form [12]. In our vision, the KRR should not only extract knowledge from data but also learn knowledge from system actions that can support the reasoning process. Knowledge reasoning can be seen as the fundamental building block that allows machines to simulate humankind's thinking and decision-making [13]. With generations of development on KRR, the current most promising approach is the knowledge graph (KG) [14] derived from the semantic web [15] community. A knowledge graph has two layers of representation structure: 1. pre-defined ontology and vocabularies and 2. instances of triple statements (e.g. dog isA Animal, where the dog is an instance, isA is a predict while Animal is a concept vocabulary defined in the ontology). The reasoning part is to apply the logical side of the ontology, such as description logic (e.g. is the dog an animal? reasoning result is 'Yes') [16]. There are many complex types of ontologies developed in the last decade to solve different KRR problems and applications. The most important development of ontology-driven reasoning is to encode dynamic uncertainty [17], probability [18] and causality [19]. Therefore, the KG-based KRR framework can be applied to implement our proposed vision.

2.4. Services and Machine Learning Ontologies

The web services community has researched auto-configuration or service composition for many years by applying a variety of dynamic integration methods. There are two trends in service composition research:

- Directly extracts the services description file (e.g. WSDL) and Quality of Services (QoS) into a mathematical model with a logical framework for composing services such as a linear logic approach [20] and genetic algorithms [21, 22]. The major limitation is that there are no formal specifications for modelling and reasoning. Therefore, the processes are mostly hard-coded to match the logic framework.

- The other trend is to apply Semantic Web standards for semantically encoding services description and their QoS properties (Semantic Web

Services SWS)[23, 24]. The main benefit is that semantic annotation has an embedded logical reasoning framework to deal with composition tasks.

On the one hand, the semantic web services (SWS) trend has greater strength for integrating the KRR approach with the same semantic infrastructure and reasoning logic. Currently, there are three standards of OWL-S (composition-oriented ontology), WSMO (task-goal matching oriented ontology), and WSDL-S (invocation-oriented ontology). On the other hand, there are two differences between our vision's microservice to normal SWS. The first one is that AIMSs have simpler input and output requirements to perform an efficient composition process. The other is that the purpose of each microservice is to deal with data analytic or machine learning tasks. Therefore, the AIMS ontology needs to be defined by modifying current machine learning ontology standards. Researchers have realised that there is a need to have a machine learning ontology and some recent proposals in this domain are: the Machine Learning Schema and Ontologies (MLSO) introduces twenty-two top-layer concepts and four categories of lower-layer vocabularies (the detailed ontology design is in [25]); the Machine Learning Ontology (MLO) proposes to describe machine learning algorithms with seven top layer concepts of Algorithm, Application, Dependencies, Dictionary, Frameworks, Involved, and MLTypes [26].

2.5. The gaps

By reviewing the current state of the art, we found that there are remaining research gaps to achieve our goal.

- Self-supervised knowledge generation during the machine learning process and solution creation. In the past, knowledge generation system mainly refers to expert systems that acquire knowledge from human expertise or systems that transform existing knowledge from one presentation to the other. [19] presents an automatic process of disease causality knowledge generation from HTML-text documents. However, it still doesn't fully address the problem of how to automatically learn valuable knowledge from the whole task-solution-evaluation machine learning life cycle. Considering human-level intelligence, we always learn either directly from problem-solving or indirectly through other human expertise (e.g. reading a book or watching a video) or a combination of bother (e.g. reflecting on the opinions of others).

- Provisioning knowledge-guided auto-ML solution. In contrast to the first gap, there are no significant research works on using the knowledge to assist in providing an AI solution. Again, compared to human-level intelligence, we always try to apply acquired knowledge or knowledge-based reasoning to solve the problem. We can consider that the transformer process [27] is a step forward in this direction. We can treat well-trained AI models as a type of knowledge to apply to different tasks in a similar problem domain. However, there is still no defined framework that can specify what knowledge is required and how to use the knowledge to find a solution to new tasks [5].

3. Self-supervised Knowledge Learning for Solution Generation

The self-supervised knowledge learning approach involves three types of auto-configuration transfer learning methods. Figure 2 presents the overall learning framework.

The first method is knowledge space searching and transferring: A task with a dataset (referred to as a task-context) arrives, and there is no previous knowledge related to the task-context. Therefore, the knowledge space will be searched to try to find a possible microservice that can match the context to complete the task or search for a pipeline (workflow) that contains multiple I/O compatible AIMSs together towards the best and successful completion which can be optimised. The task-context and the optimized solution are recorded as task input and output knowledge. The evaluation will generate rewards for the

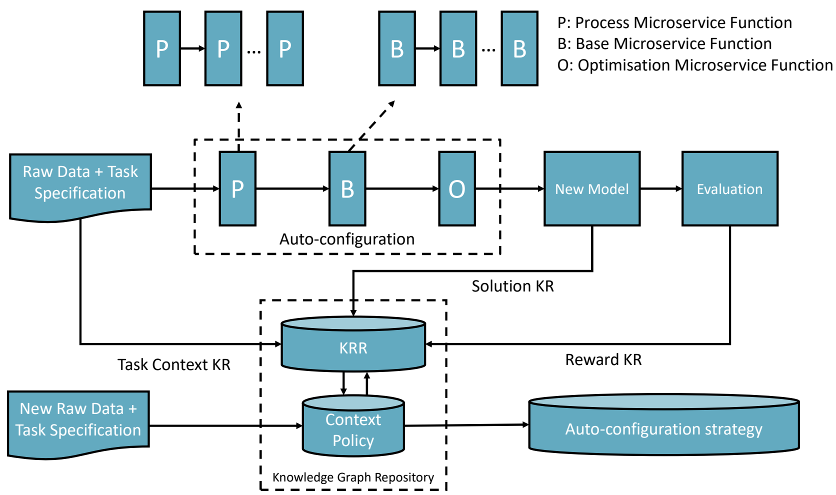

Figure 2: The vision of the self-knowledge learning approach with AI Microservices

policy knowledge space. In addition, the knowledge learnt from the process will be recorded to update the world knowledge space.

The second one is the mutation of a previously generated context-matched solution. If the new task-context matches with a previously recorded task-context in the knowledge space, then the previous solution will be loaded to adapt to the new datasets and the optimisation process. Finally, a new mutated solution is created and recorded as new knowledge with the new evaluation rewards and world knowledge of the KRR environment.

The third one is the continuous learning mutation method based on the reinforcement learning approach. With the growth of the KRR statements, the auto-mutation will take place using world knowledge to re-train the solutions according to the rewards. The third learning method takes place offline only but continues doing an update when KRR is updated.

4. Experimental Implementation

4.1. Overview of the implementation structure

Figure 3 shows a three-layer implementation of the vision. It also shows how these layers map to the human-level AI process.

- **Request layer** takes tasks and inputs from AI applications to trigger the solution searching and self-learning processes. Task-context is semantically encoded to enable starting the policy knowledge to explore the environment for learning, creating, or finding solutions.

- **Reasoning layer** takes the request layer's semantic reasoning tasks for semantic matching, reasoning and doing reinforcement learning mechanism. Finally, the policy will be recorded in the knowledge graph layer. In addition, the newly added AIMSs are registered to the environment with semantic annotations through knowledge registration and generation components.

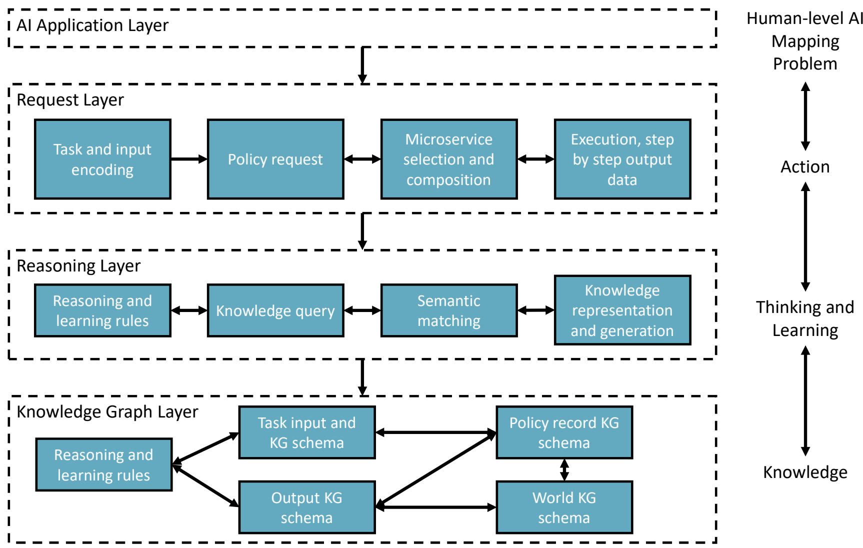

Figure 3: Self-supervised Knowledge Learning and Reasoning Framework Design

- **Knowledge Graph layer** remembers the knowledge data in the knowledge graph triple store based on different types of knowledge schema.

4.2. Knowledge ontology implementation

4.2.1. AI microservice knowledge modelling

AIMS registration ontology defines 9 parameters (see Figure 4).

- *name* - must be no duplication in the system, and the registration process will check the name's legibility.
- *description* - a short presentation of the AIMS for human understanding.
- *framework* - indicates the programming framework used to develop the AIMS. Normally, it should be just one framework as AIMS is designed to be decoupled and ideally single responsibility.

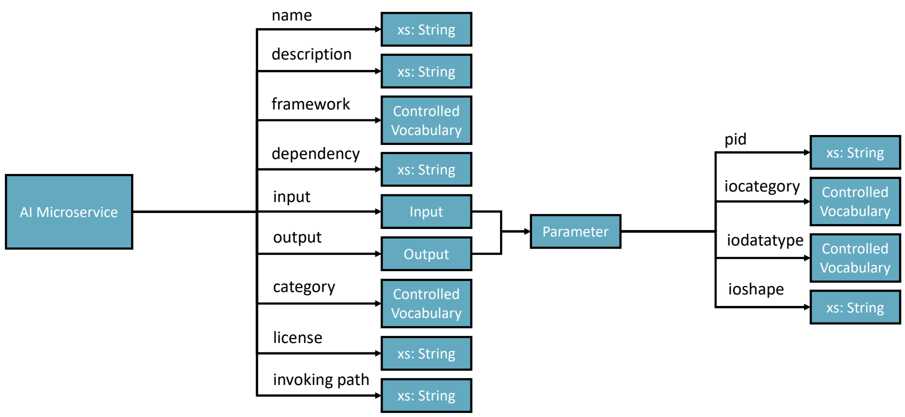

Figure 4: Self-supervised Knowledge Learning and Reasoning Framework Design

- *dependency* - describes the required programming libraries that need to be pre-configured to enable the AIMS to work.
- *input and output* - specifies the parameters that should be in the input and output messages.
- *category* - tells what AI-related domain the ms works on, such as supervised classification, unsupervised clustering, image classification base model, and more.
- *license* - identifies the use conditions and copyright of the AIMS.
- *invoke path* - contains the portal for accessing the AIMS. The path can be a local path or URI of a restful API.

4.2.2. Task-context knowledge modelling

Each given task triggers a context knowledge creation that collects the knowledge of:

- *the type of input data* - a controlled variable that majorly includes normal dataset (e.g. CSV file), image, and text.
- *task domain* - free text to record the specific application domain
- *desire output type* - records the output required to complete the task successfully.
- *the features* - the dataset or data presents the initial characters of the input data. For example, the number of columns and column names will be remembered as part of the context knowledge of the task.

4.2.3. Policy knowledge modeling

The output of the performed task can be categorised into two types failure and success. Both failure and success need to update the policy knowledge link to the task-input context. Failure has no solution registered to the knowledge but records which AIMSs have been successfully invoked (can be an empty list) until the step that cannot continue going further. So the failure experience will tell the system administrators (people) what AIMS(s) are required to create a solution. The success registers the solution location and changes the policy with the reward value. If the solution contains a workflow of composed AIMSs, then the workflow will also be registered as knowledge with the normalized rewards for each of the AIMS.

The ontology was designed as:

- *policy context* - links to a task-context
- *policy state* - 1 is success and 0 is failure
- *solution iloc* - the location where the solution can be loaded and executed.
- *workflow* - presents a pipeline solution that composes multiple AIMSs.

- *solution reward* - the reward value stored for the policy that can be the recommended guidance for supporting the creation of a new task solution.

4.2.4. World knowledge modeling

The world knowledge presents the facts learned from the task solution creation process and outputs. There are three types of world knowledge recorded in the current environment:

- *feature optimisation outcomes* - the features selected in the optimisation are valuable, and these features will be reused to create a classification model if the new dataset features are the same.
- *answers for a certain text topic* - a generated text answer for a question. The answer quality will be reported as a reward value feedback from humans back to the policy knowledge.
- *image RBG vectors* - map to a classification label. The reward process is the same as the answers.

More world knowledge can be expanded in the environment. By having these commonsense and policy records, reinforcement can be performed to improve the solution accuracy incrementally.

4.3. Environment initialization

The experiment environment is developed by Python in a local single-computer environment. We simplified the AIMS as a .py module in the environment to be invoked and registered. The full implementation can be found in GitHub repository (`https://github.com/semanticmachinelearning/AISMK`). We initialised the environment with three types of AIMS

1. Data Processing AIMSs that include CSV file to a Panda service, Text tokenisation Service, Image reading Service, Data Split service

2. ML AIMSs that include classification services, GPT-neo-1.3B text generation services [28, 29], ViT image classification transformers [30]

3. RFECV optimisation service

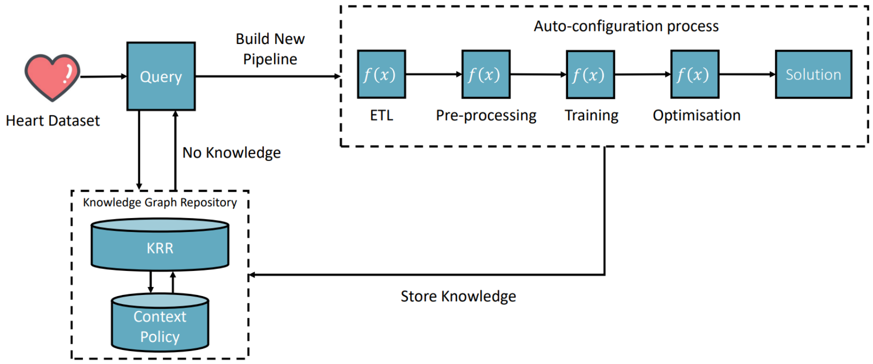

Figure 5: Scenario 1 - Heart disease classification solution building and knowledge learning process

5. Scenario Evaluation and Lesson Learned

5.1. Heart disease classification scenario

Figure 5 presents one of our use case scenarios in the medical domain. The task context is:

- *input*: 335 clinical CSV heart disease files labelled 0 (no disease) and 1 and 1 (confirmed disease)
- *domain*: medical
- *desire output*: an optimised classification pipeline model

The application domain is medical and the desired output is an optimised classification pipeline model. Figure 5 illustrates how the process starts by searching existing knowledge (no knowledge found) and creates a workflow containing four AI microservices for loading the CSV data, splitting the data, classification pipeline creation, and optimisation to get the best accuracy model. Different pieces of knowledge are learned and recorded in the system environment.

5.2. Parkinson disease classification scenario

The second task-context is:

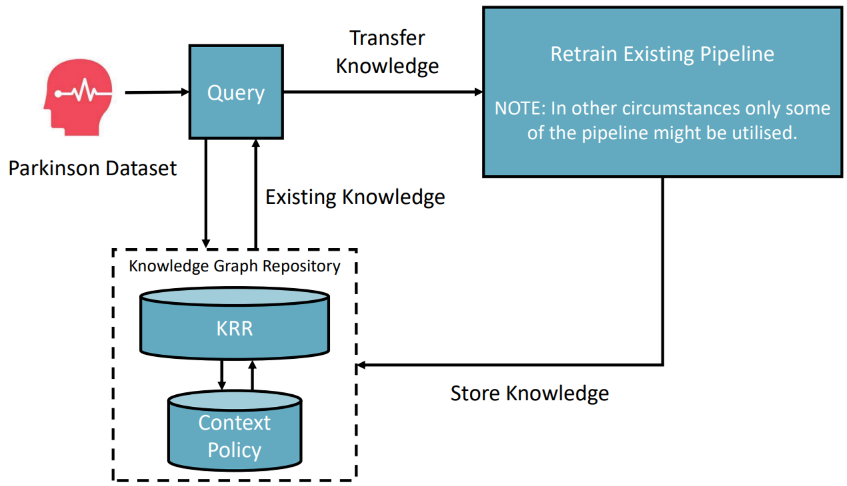

Figure 6: Scenario 2 - Parkinson disease pipeline transfer classification process

- *input*: csv Parkinson disease clinical example data with labelled 0 (no disease) and 1 (confirmed disease)
- *domain*: medical
- *desired output*: an optimised classification pipeline model

Figure 6 depicts the scenario in which a similar task of classifying Parkinson's disease is fed in, the framework starts searching for a solution. As the system environment has pre-knowledge, gained through the previous heart disease classification, and since the only difference is the dataset, the framework can use the classification pipeline and retrain it to be optimised for the new dataset. We can call this process a composition transfer learning process. The novelty is that the system environment can solve different tasks by applying contextual knowledge of the problem. Thus, the framework can automatically deal with all types of data if the required models are semantically registered in the framework.

5.3. Image to text classification scenario

The third task context is:

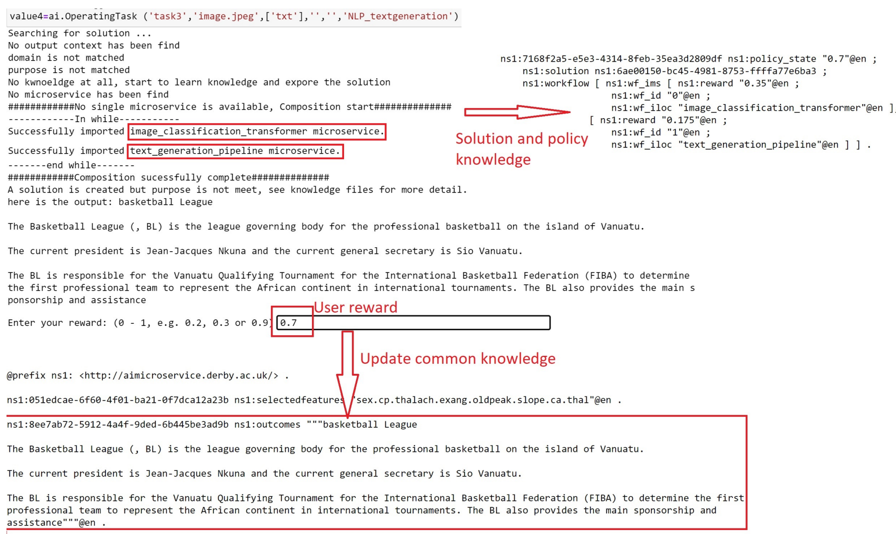

Figure 7: Scenario 3 - Generating text based on an image input

- *input*: an image (jpeg) file
- desire output: the text contents that relate to the image

If this is the first time a task takes an image as input, then the system environment does not have any pre-knowledge. The system will then search for appropriate AIMSs and then invoke these to successfully create a solution. After a solution is generated, the user can then provide a reward value to rank the output. This reward is then stored in the knowledge space. The output is also recorded in the common knowledge space links to the policy context that stores reward values. Hence over time, the system is being educated by the human user. If the system sees similar input in the future, it can use the acquired knowledge to provide better output to the user. An example is given in Figure 7.

5.4. Text generation improving scenario

The last task context is:

- *input*: keyword text

Figure 8: Scenairo 4 - Learning the same topic to generate text

- *desire output*: generated text content related to the keywords

The keywords "basketball league" can be given to the system to start generating related text content. If there are enough learning cases related to the same keywords, then the system can provide the best output faster based on the historical ranking value (see Figure 8). This process can reduce the computational cost by not producing new content via a large text generation model (e.g. GTP). In addition, the updated knowledge can also improve the outputs of the third task scenario.

5.5. Lesson learned

By evaluating the performance of the test scenarios, we believe the combination of KRR and automation of AIMSs is one possible approach to developing human-level AI systems. We implement the environment that has AIMSs that can take text, CSV files, and images as default settings with splitting data, classification, prediction, and optimisation AIMSs. It shows that system can generate and optimise solutions for different types of tasks by applying or creating knowledge. However, there are some limitations that need to be addressed in future work:

- The benefit of applying a triple KG structure to encode KRR elements is the unification, standardization, and ease of adoption by different applications. However, if the KG grows too big, the efficiency of referencing based on the complex graph query is very slow. Especially, if the different types of knowledge are stored separately, then the union querying on the graph is very costly. The future work direction is to efficiently embed the Knowledge Graph into a more efficient vector space [31].

- The current implementation cannot take multiple inputs that belong to one task to generate a solution (Multi-modal Machine Learning). Humans can take images, data, sound, and smell together to complete the task, but it is still a challenge for the machine to merge different types of inputs together to solve the problem [32, 33, 34].

- When we started writing this paper, the Google research papers were published introducing the mutation of Neural Network (NN) to deal with multiple image classification tasks [5, 6]. This latest paper provides some new ideas to us that not only data-based mutation but also actions of inserting or removing NN hidden layers can be stored as knowledge in the future.

6. Conclusion

The long-term goal of AI research is to achieve something akin to human-level thinking. The hypothesis of this research paper is that the machine self-knowledge learning process can be performed to support the automation of a semantic AIMS solution that provisions for multiple types of tasks and data inputs. Therefore, the whole idea can be seen as a knowledge-led problem-solving approach, which is one step forward to achieving human-level intelligence. The evaluated scenarios proved that our hypothesis is valid, although there are some open limitations to be addressed in our future work.

References

[1] A. M. TURING, I.—computing machinery and intelligence, Mind LIX (236) (1950) 433–460. arXiv:https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf, doi:10.1093/mind/LIX.236.433.
URL https://doi.org/10.1093/mind/LIX.236.433

[2] D. Park, Toward human-level artificial intelligence, CoRR abs/2108.03793. arXiv:2108.03793.
URL https://arxiv.org/abs/2108.03793

[3] M. Färber, F. Bartscherer, C. Menne, A. Rettinger, Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO, Semantic Web Journal (2017) 1–53.

[4] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, I. Kompatsiaris, K. Kinder-Kurlanda, C. Wagner, F. Karimi, M. Fernandez, H. Alani, B. Berendt, T. Kruegel, C. Heinze, K. Broelemann, G. Kasneci, T. Tiropanis, S. Staab, Bias in data-driven artificial intelligence systems—an introductory survey, WIREs Data Mining and Knowledge Discovery 10 (3) (2020) e1356. arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1356, doi:https://doi.org/10.1002/widm.1356.
URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1356

[5] A. Gesmundo, J. Dean, An evolutionary approach to dynamic introduction of tasks in large-scale multitask learning systems (2022). doi:10.48550/ARXIV.2205.12755.
URL https://arxiv.org/abs/2205.12755

[6] A. Gesmundo, J. Dean, munet: Evolving pretrained deep neural networks into scalable auto-tuning multitask systems (May 2022). doi:10.48550/

ARXIV.2205.10937.
URL https://arxiv.org/abs/2205.10937

[7] Y. LeCun, A path towards autonomous machine intelligence (June 2022).

[8] Q. Yao, M. Wang, H. J. Escalante, I. Guyon, Y. Hu, Y. Li, W. Tu, Q. Yang, Y. Yu, Taking human out of learning applications: A survey on automated machine learning, CoRR abs/1810.13306. arXiv:1810.13306.
URL http://arxiv.org/abs/1810.13306

[9] H. Jin, Q. Song, X. Hu, Auto-keras: An efficient neural architecture search system (2018). doi:10.48550/ARXIV.1806.10282.
URL https://arxiv.org/abs/1806.10282

[10] L. Sharma, P. K. Garg, Knowledge representation in artificial intelligence: an overview, Artificial intelligence (2021) 19–28.

[11] F. G. Cozman, H. N. Munhoz, Some thoughts on knowledge-enhanced machine learning, International Journal of Approximate Reasoning 136 (2021) 308–324. doi:https://doi.org/10.1016/j.ijar.2021.06.003.
URL https://www.sciencedirect.com/science/article/pii/S0888613X21000840

[12] Z. Hu, Z. Yang, R. Salakhutdinov, E. Xing, Deep neural networks with massive learned knowledge, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1670–1679.

[13] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Systems with Applications 141 (2020) 112948. doi:https://doi.org/10.1016/j.eswa.2019.112948.
URL https://www.sciencedirect.com/science/article/pii/S0957417419306669

[14] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Systems with Applications 141 (2020) 112948.

[15] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific american 284 (5) (2001) 34–43.

[16] F. Baader, I. Horrocks, C. Lutz, U. Sattler, Introduction to description logic, Cambridge University Press, 2017.

[17] Q. Zhang, Dynamic uncertain causality graph for knowledge representation and probabilistic reasoning: Directed cyclic graph and joint probability distribution, IEEE Transactions on Neural Networks and Learning Systems 26 (7) (2015) 1503–1517. doi:10.1109/TNNLS.2015.2402162.

[18] L. BOTHA, T. MEYER, R. PEÑALOZA, The probabilistic description logic, Theory and Practice of Logic Programming 21 (4) (2021) 404–427. doi:10.1017/S1471068420000460.

[19] H. Q. Yu, S. Reiff-Marganiec, Learning disease causality knowledge from the web of health data, International Journal on Semantic Web and Information Systems (IJSWIS) 18 (1) (2022) 1–19. doi:10.4018/IJSWIS.297145.

[20] X. Zhao, E. Liu, H. Q. Yu, G. J. Clapworthy, A linear logic approach to the composition of restful web services 10 (3) (2015) 245–271. doi:10.1504/IJWET.2015.072348.
URL https://doi.org/10.1504/IJWET.2015.072348

[21] M. Allameh Amiri, H. Serajzadeh, Qos aware web service composition based on genetic algorithm, in: 2010 5th International Symposium on Telecommunications, 2010, pp. 502–507. doi:10.1109/ISTEL.2010.5734077.

[22] B. Qiang, Z. Liu, Y. Wang, W. Xie, S. Xina, Z. Zhao, Service composition based on improved genetic algorithm and analytical hierarchy process, International Journal of Robotics and Automation 33. doi:10.2316/Journal.206.2018.2.206-5106.

[23] H. Q. Yu, X. Zhao, S. Reiff-Marganiec, J. Domingue, Linked context: A linked data approach to personalised service provisioning, in: C. A. Goble, P. P. Chen, J. Zhang (Eds.), 2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012, IEEE Computer Society, 2012, pp. 376–383. `doi:10.1109/ICWS.2012.16`.
URL `https://doi.org/10.1109/ICWS.2012.16`

[24] H. Dong, F. Hussain, E. Chang, Semantic web service matchmakers: State of the art and challenges, Concurrency and Computation: Practice and Experience 25. `doi:10.1002/cpe.2886`.

[25] G. C. Publio, D. Esteves, A. Lawrynowicz, P. Panov, L. N. Soldatova, T. Soru, J. Vanschoren, H. Zafar, Ml-schema: Exposing the semantics of machine learning with schemas and ontologies, ArXiv abs/1807.05351.

[26] J. Braga, J. Dias, F. Regateiro, A machine learning ontology, Frenxiv Papers`doi:10.31226/osf.io/rc954`.

[27] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2018, Springer International Publishing, 2018, pp. 270–279.

[28] S. Black, G. Leo, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, If you use this software, please cite it using these metadata. (Mar. 2021). `doi:10.5281/zenodo.5297715`.
URL `https://doi.org/10.5281/zenodo.5297715`

[29] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al., The pile: An 800gb dataset of diverse text for language modeling, arXiv preprint arXiv:2101.00027.

[30] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, A. Veit, Understanding robustness of transformers for image classification, in: Pro-

ceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 10231–10241.

[31] T. Le, N. Huynh, B. Le, Link prediction on knowledge graph by rotation embedding on the hyperplane in the complex vector space, in: I. Farkaš, P. Masulli, S. Otte, S. Wermter (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2021, Springer International Publishing, Cham, 2021, pp. 164–175.

[32] D. Ramachandram, G. W. Taylor, Deep multimodal learning: A survey on recent advances and trends, IEEE signal processing magazine 34 (2017) 96–108.

[33] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, IEEE transactions on pattern analysis and machine intelligence 41 (2018) 423–443.

[34] J. Li, D. Hong, L. Gao, J. Yao, K. Zheng, B. Zhang, J. Chanussot, Deep learning in multimodal remote sensing data fusion: A comprehensive review, arXiv preprint arXiv:2205.01380.