Neuro-symbolic A.I. for the smart city

Gilles MOREL
Alliance Sorbonne University, Université de Technologie de Compiègne, EA 7284 Avenues, Centre Pierre Guillaumat, 60203 Compiègne, France
gilles.morel@utc.fr

Abstract. Smart building and smart city specialists agree that complex, innovative use cases, especially those using cross-domain and multi-source data, need to make use of Artificial Intelligence (AI). However, today’s AI mainly concerns machine learning and artificial neural networks (deep learning), whereas the first forty years of the discipline (the last decades of the 20th century) were essentially focused on a knowledge-based approach, which is still relevant today for some tasks. In this article we advocate a merging of these two AI trends – an approach known as neuro-symbolic AI – for the smart city, and point the way towards a complete integration of the two technologies, compatible with standard software.

1. Introduction
This article offers a global reflection and puts forward a proposal regarding the application of Artificial Intelligence (AI) theory and technology for the smart city and the smart building. We consider the synergy and integration of the two historical approaches and technologies in AI, both at the conceptual and operational (software) levels. The connectionist approach is represented today by machine and deep learning, while the symbolic, knowledge-based approach that prevailed during the second half of the 20th century was mediatized in industry and for the public at large in the shape of expert systems.

The city and the building are complex systems, and smart applications claiming to address issues of sustainability, climate change, and the circular economy need to move beyond a single-domain view and to harness multi-domain and multi-source data and knowledge. The processing of this kind of heterogeneous data for complex problem solving cannot easily be done by a single model, nor even by a single family of models.

For this reason, we recommend, for complex applications that potentially fall into the sphere of AI, the development of methods and tools based on a neuro-symbolic approach, that is to say on a merging of machine learning (including artificial neural networks) and knowledge-based systems.

2. Urban big data processing for complex issues: which models?

2.1. New services for the smart and sustainable city
For two decades the development and deployment, especially in built-up areas, of powerful communication networks and cheap, lightweight devices such as sensors and smartphones has been making data acquisition and transmission easier, leading to the emergence of urban big data. The same period has seen a profusion of innovative digital services in the private and professional spheres, often deriving from and making use of these new data. The context of the thoughts and proposals presented in this article encompasses complex use cases of urban systems including transport (traffic
management), water supply, energy, air quality, construction, and eco-districts. Over the last decade a lot of these innovations have moved beyond the perimeter of a single business specialty (often called a “silo”) and have begun to merge data from different domains, sources and types, as described in [1].

Smart city applications that make use of multi-source and multi-domain data include traffic forecasting [2], air quality forecasting [3], and the detection of anomalies in energy consumption [4]. Most urban and building applications developed over the last ten years and presented in the scientific literature employ machine learning (including deep learning), generally in relation to data from sensor networks. As we describe below, this kind of machine learning is the dominant trend in AI today.

2.2. The origin and nature of urban data
The urban data that can be used by new services for the smart city – and also for the smart building – can come from different sources, the main categories being (1) sensor networks (time series); (2) geospatial – or 3D – databases; (3) crowdsourcing: data from connected people and from the field; and (4) information provided by experts in the area of interest. These sources can be useful and complement each other in analysing and diagnosing a complex urban system, potentially covering the five main “dimensions” of urban data: (a) spatial, (b) temporal, (c) contextual, (d) domain-oriented, and (e) relational and causal. In this paper we cannot provide a complete review and classification of urban data, but we refer the reader to [5]. Ideally, effective problem solving and data processing call for the merging of all these heterogeneous into a single, coherent data model, based on an ontology of the domain (the set of concepts and their relations). Unfortunately, existing standard tools – and their associated databases – that target the building and the city (see the next paragraph) fail to cover all of these dimensions, their main focus being the spatial and the domain-oriented.

2.3. Compatibility with standard urban and building tools
For describing the structural and technical characteristics of a building or a city, some existing general models and tools can be considered to be international de facto standards. These include the Building Information Model (BIM, in 3D), and Geographical Information Systems at the urban scale (GIS, mainly in 2D). These converge in the City Information Model (CIM, the city in 3D). When AI technology is applied in relation to the smart city and the smart building, there is a tendency for these existing tools to be ignored and for a city model to be developed “from scratch”. However, as we have mentioned above, these existing tools mainly describe the spatial (and geometrical) dimension, together with domain-related data (technical characteristics), but fail to address other dimensions adequately, especially the temporal dimension, causal relations, high-level and multi-domain analysis functions. Now, when developing high-level reasoning models using AI, there exist a variety of techniques (which we cannot describe in detail here) for coupling an AI platform with the BIM and GIS standards. One technique consists in developing or using a middleware layer (like CORBA – [6]) to encapsulate the classes and libraries of the different modules, with a shared API (application programming interface).

2.4. The limits of a single modelling approach
Apart from the question of the data model (or the domain model), a single problem-solving method is generally unable to address the sort of complex needs that arise within the smart city, such as building a map of forecasted air quality levels from multi-source and multi-domain information. To illustrate this, we will discuss further the limitations inherent in a machine learning process.

In the literature there are a number of research works concerning complex smart city and smart building use cases that propose a multi-model approach and coupling, based either on a sequential or “pipeline” approach (typically in the machine learning process) [7], or on an approach involving some kind of concurrency or comparison between models [3]. These studies often suffer from two drawbacks: first, they are linked to a specific use case and cannot easily be generalized for different cross-domain applications, and secondly the different models invoked often belong to the same family of models (for example, they attempt to compare or combine different techniques of machine learning). In contrast, the
arguments and proposals that we develop in the next two sections concern the combining of two opposing approaches and models representing the two main trends in AI.

3. AI complementary approaches and techniques

3.1. Connectionist learning vs symbolic reasoning

In order to better understand the origin and nature of the neuro-symbolic approach and its interest for smart city applications, it is necessary to briefly recall the history of AI and its two main branches. Since the emergence of AI as an academic field in the 1950s, there have been two main theoretical approaches and technologies in the attempt to reproduce or mimic human intelligence and reasoning. The first decades of AI were dominated by the semantic and symbolic approach (mediatized to the public through expert-systems), whereas today there is a far greater focus on the connectionist approach, implemented using machine/deep learning methods and tools (artificial neural networks – ANNs – for deep learning).

The connectionist approach, whose objective is to reproduce the structure and operation of the human brain (with its neurons, synapses, signal processing, and so on) was already present in the early years of AI (the perceptron, Rosenblatt, 1957), but the technical conditions for an effective implementation of ANNs, and of machine learning models more generally, were not available at that time, and have been achieved only recently, requiring the availability of big data, adapted algorithms to calibrate and train the ANN (backpropagation, 1986), and a tremendous computational capacity provided by CPUs/GPUs, cloud computing, etc.

3.2. Strengths and limits of machine learning and ANNs

Machine learning (ML) involves calibrating and “training” an existing model and algorithm (ANN in the case of deep learning, which is a special case of ML) on a sample of validated data including inputs and output(s), the latter being the final target or solution of the problem. Once the model is trained and adjusted with these data, it can be given a new set of input data for it to “guess” or predict the corresponding output(s), which can be a qualitative value (such as a diagnosis or a classification), or a quantitative value (such as a prediction of the level of particulate matter over 24 hours at an urban location). Most ML models (except ANN) work well on medium-sized, structured samples of data, whereas ANNs are more suitable for large quantities of unstructured or semi-structured data (such as in image analysis and natural language processing).

In the field of urban systems and smart buildings, these techniques have been applied above all to time-series produced by sensors in order to detect anomalies, predict future values from the past and optimize complex systems and issues like traffic [2], air quality [3], energy consumption [4], and activity recognition [8]. Real use cases are quite complex, because heterogeneous and multi-source datasets need to be analysed, which requires multivariate time-series analysis models like VAR (statistic-oriented ML) or LSTM (a variant of recurrent neural networks) [2].

The success of ML and ANNs can be explained by the apparent simplicity of using “off the shelf” libraries of models with existing data, avoiding the need to develop a specific model of the domain and the targeted use case. Nevertheless, real use cases developed in the scientific literature and on specialized websites like Kaggle\(^1\) show that the actual process of a machine learning project is complex and requires real expertise if relevant results are to be obtained. The project will involve cleaning, visualizing, scaling the data, extracting the relevant features, choosing and training the appropriate model and tuning its parameters, evaluating the results, restarting the whole process for the purposes of optimization, and so on. In contrast to the knowledge-based approach that we describe below, the expertise required in handling the model is not explicitly integrated inside the model and in the code, which has consequences and drawbacks. For a start, this expertise, which draws on both machine learning know-how and knowledge of the relevant field, is not automated and not wholly reproducible from one study to another.

\(^1\)https://www.kaggle.com/
Moreover, the results produced by the model are not explainable, since the model is a “black box” employing an obscure mathematical algorithm. This is a real problem for some decision support and critical applications, where the diagnosis produced needs to be understood and explained. The strengths and weaknesses of machine learning, which can be seen as a data-driven, numerical orientation within AI, are precisely the inverse of those that we encounter in symbolic AI. As we shall now describe, symbolic AI is particularly successful in addressing aspects that are problematic for machine learning, and vice versa.

3.3. Strength and limits of symbolic AI and knowledge base tools

The approach that was dominant in AI up to the 1990s rests on the hypothesis that human reasoning can be made completely explicit and expressed first using semantic primitives (concepts of the domain, expert rules, resolution strategy …), then with symbolic representations, and finally implemented via an ad-hoc computing language like LISP or PROLOG. Developing applications with this approach is very different from using machine learning. Not much data is needed to establish the model, but significant human effort is required to formalize and develop the knowledge base. However, once the latter has been validated, the expertise is supposed to be present in the code, and the reasoning reproducible and explainable. This approach is suitable when data are sparse (another difference with machine learning) and when the domain-specific knowledge (concepts and relations) and the reasoning process (the heuristic rules, the tasks …) can be made explicit. Symbolic AI has given rise to several types of models and languages, and is the basis of expert systems, which have had some success in industry and in areas including medicine and engineering. One drawback of this approach is the cost of maintaining and adapting large knowledge bases over the long term. Concepts and tools relating to smart cities and smart buildings have appeared in the context of machine learning, but have not really benefited from the experience and feedback that we might expect with expert systems (hence the lack of examples and reference in the literature). However, introducing domain expertise in the modelling process and in the operational model itself would seem useful for a majority of applications. In the next paragraph we present the interest of combining machine learning and symbolic AI to get the best out of the two approaches and to overcome some of their respective shortcomings.

4. A neuro-symbolic AI approach for urban computing

4.1. Why neuro-symbolic AI is necessary and powerful

Both of the two main trends in AI presented above are encompassed by cognitive science, and both reflect a way that humans actually think and solve problems in everyday and professional life. Connectionist AI, represented by machine and deep learning, expresses the capacity to process a large quantity of information – typically an image – unconsciously, without elaborating a rational reasoning, such as when recognizing a cat or a person in a static or dynamic scene. This kind of tool would appear necessary in a smart city to quickly analyse all the images, videos and other big data produced by sensors and other devices. In contrast, designing a car, a plane or diagnosing a complex (urban) system calls for a conscious manipulation of explicit concepts and rules in a rational way: this corresponds to the symbolic, knowledge-based approach in AI.

It is difficult to imagine a human being deprived of either of these essential abilities for tackling daily challenges and complex problems, and this helps explain why using only one AI technology gives only limited results and sometimes leads to a dead-end. Coupling the capacity of knowledge bases and neural networks (and/or machine learning) is called “neuro-symbolic AI”. Surprisingly, this is an area of research in its own right, which has existed for decades but which was until recently “under the radar” and neglected by the scientific community. In the coming decade neuro-symbolic AI is expected to become one of the major axes of AI research. Some of the most remarkable results obtained by AI in recent years have been achieved thanks to tools that have already integrated this dual approach. Examples are Watson from IBM, which first won the Jeopardy TV game in 2011, and AlphaGo from Google DeepMind, which defeated the GO world champion in 2015.
4.2. How to make the two kinds of AI collaborate

Once it is accepted that both machine learning and knowledge-based systems are complementary in solving complex problems of the smart city, the question is how precisely to couple the two approaches and technologies. A fairly comprehensive recent overview of this topic can be found in [9]. At least six main categories of coupling are identified, although some of these have a targeted and limited objective, such as transforming one representation into another, for example a symbolic representation into an ANN, or vice versa. Where simple association rules are extracted from a machine learning process, they can be used to explain the outputs of a prediction, thus producing an “explainable AI”. However, results in this area remain largely unsatisfactory in terms of symbolic knowledge expression, and generally leave the main body of the task to the machine learning process, which is equivalent to admitting that in the final analysis the unconscious mind takes precedence over rational decision making.

The authors of [9] conclude that an architecture and language for combining the two AI technologies effectively does not exist today. Current reflection on this subject is a little distorted by the fact that machine learning and ANNs are at present the dominant technology. The culture of knowledge-based AI has lain dormant for thirty years and has largely been lost by the community. Two vital components are missing for a fully, stable integration of the two AIs: (1) an AI software architecture designed at a conceptual level, and (2) a high-level knowledge-based language compatible with industrial standards.

4.3. The contribution of knowledge engineering as a conceptual framework

The criticism of first generation expert systems in the 1980s contributed to the emergence of knowledge engineering and methods like CommonKads [10], which clarify the nature and status of different types of knowledge at the conceptual level (i.e., independently of symbolic representations and languages). The three main categories of knowledge are the domain, the problem solving and the control. In particular, the problem-solving layer distinguishes the notion of a task (linked to a goal) and the notion of a problem-solving method, which refers to the way a task is implemented (possibly with different models). All this may appear very theoretical, but it should facilitate the design of a high-level language for the purpose of integrating and piloting a multi-model toolbox that includes both machine learning algorithms and symbolic reasoning methods (logic and heuristic rules), and possibly other kinds of models (such as graphs and causal models). Taking an example from the city, a goal and task might be “Find the correlations and causal links between different sensors in a district”, and this task could mobilize and provoke cooperation between different methods of data analysis (machine learning) and/or expert analysis (possibly expressed in terms of predicate logic, heuristic rules, or fuzzy logic).

4.4. A core implementation with frames in Python

The second problem is the lack of a high-level knowledge representation language (KRL), compatible with today’s software standards. The most advanced KRLs (such as KL-ONE [11]) were developed in LISP in the 1980s, but are no longer used industry. Attempts to adapt these languages together with some expert system tools, using predominantly C++ in the 1990s, and subsequently Java, were not convincing, partly because these popular languages lack certain characteristics needed for symbolic AI and that natively part of LISP features: functional programming, dynamic typing, interpretability, access to the object meta-level, symbolic and list processing.

Fortunately, the Python language has incorporated those characteristics, and would be a good candidate to transfer and adapt the best AI symbolic languages, with a view to complete interoperability with other tools like machine learning libraries, GIS, and BIM. Most, if not all, of the models and algorithms of machine and deep learning are nowadays available as Python libraries (Scikit-learn, Keras, etc.) and have been implemented via a very effective, modular object-oriented design, thus facilitating their reuse and integration in a higher level AI architecture based on the CommonKads knowledge level.

The conceptual and knowledge primitives cannot, however, be efficiently implemented directly in a basic object language like Python. An intermediate symbolic structure and language is needed, such as the frame, a theory and language representation proposed and developed by Marvin Minsky at MIT in the 1970s [12]. Frames are a pragmatic AI approach – unlike formal logic – that proved to be well
adapted to addressing technical and engineering problems. A frame represents a situation typically encountered in the world, that can include physical objects but also abstract concepts and reasoning structures. This theory and technology belongs to the pragmatic branch of symbolic AI and has given rise to a set of AI languages, including operational languages used for second-generation expert systems [13]. A frame language can be implemented with a dynamic object-oriented language like Python, and can also encapsulate different types of problem-solving methods, irrespective of their nature and origin. This proposal will be developed in a future article.

5. Conclusions and perspectives
The city and the building are complex systems, and developing new AI applications for urban systems calls for the merging of a data-oriented approach (machine learning and ANN) with knowledge-based systems, and possibly other types of models.

With this objective, we argue that a neuro-symbolic AI software architecture able to encapsulate all kinds of models and libraries should be based on (1) knowledge primitives at the conceptual level, (2) a knowledge representation language (resembling frames) for the symbolic level, and (3) an implementation of this knowledge representation language in Python, which is the best popular LISP-like language that is compatible with symbolic AI, machine learning libraries and industrial standards.

References
[1] LAU, Billy Pik Lik, MARAKKALAGE, Sumudu Hasala, ZHOU, Yuren, et al. A survey of data fusion in smart city applications. Information Fusion, 2019, vol. 52, p. 357-374.
[2] SCHIMBINSCHI, Florin, NGUYEN, Xuan Vinh, BAILEY, James, et al. Traffic forecasting in complex urban networks: Leveraging big data and machine learning. In : 2015 IEEE international conference on big data (big data). IEEE, 2015. p. 1019-1024.
[3] HSIEH, Hsun-Ping, LIN, Shou-De, et ZHENG, Yu. Inferring air quality for station location recommendation based on urban big data. In : Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015. p. 437-446.
[4] ARAYA, Daniel B., GROLINGER, Katarina, ELYAMANY, Hany F., et al. An ensemble learning framework for anomaly detection in building energy consumption. Energy and Buildings, 2017, vol. 144, p. 191-206.
[5] ZHENG, Yu, CAPRA, Licia, WOLFSON, Ouri, et al. Urban computing: concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 2014, vol. 5, no 3, p. 1-55.
[6] HAMILTON, Andy, WANG, Hongxia, TANYER, Ali Murat, et al. Urban information model for city planning. Journal of Information Technology in Construction (ITCon), 2005, vol. 10, no 6, p. 55-67.
[7] XIAO, Fu et FAN, Cheng. Data mining in building automation system for improving building operational performance. Energy and buildings, 2014, vol. 75, p. 109-118.
[8] QOLOMANY, Basheer, AL-FUQAHA, Ala, GUPTA, Ajay, et al. Leveraging machine learning and big data for smart buildings: A comprehensive survey. IEEE Access, 2019, vol. 7, p. 90316-90356.
[9] GARCEZ, Artur d'Avila et LAMB, Luis C. Neurosymbolic AI: The 3rd Wave. arXiv preprint arXiv:2012.05876, 2020.
[10] SCHREIBER, A. Th, SCHREIBER, Guus, AKKERMANS, Hans, et al. Knowledge engineering and management: the CommonKADS methodology. MIT press, 2000.
[11] BRACHMAN, Ronald J. et SCHMOLZE, James G. An overview of the KL-ONE knowledge representation system. Readings in artificial intelligence and databases, 1989, p. 207-230.
[12] MINSKY, Marvin. A framework for representing knowledge. de Gruyter, 2019.
[13] NEVEU, Bertrand et HAREN, Pierre. Smeci: an expert system for civil engineering design. In : Applications of Artificial Intelligence in Engineering Problems. Springer, Berlin, Heidelberg, 1986, p. 317-326.