



AI agents

Description

The sociologist Bruno Latour (1947-2022) and others who subscribe to *actor network theory* (ANT) are keen to admit *objects* and *things* as well as people into the field of social study. Hammers, kettles, baskets and remote controls [71] are participants, actors and *agents* in networks of relations. As well as providing settings for human actions, “things might authorize, allow, afford, encourage, permit, suggest, influence, block, render possible, forbid” certain actions [72]. That’s not to ascribe to inanimate things those other vital aspects of human agency to do with responsibility, liability, desire, will, morality, intention, or consciousness.

Computer programs and devices make object-based agency explicit. Though human actors have an undoubted role as they create and deploy computer algorithms and devices, there’s the possibility that such objects perform individually and collectively as unscripted actors with apparent autonomy and influence. Whatever it says about human agency, this broad definition of agency entitles us to think of natural language processing (NLP) components such as *attention* mechanisms as *agents*.

NLP attention heads as agents

The multi-head attention capabilities of NLP models such as in ChatGPT operate as collections of independent agents that create alternative patterns of attention scores across blocks of text.

These agents calculate attention matrices to weigh the importance of different parts of input data relative to each other. The multiple *heads* in the attention mechanism operate independently of one another to focus on different aspects of the data. The model integrates the information from the various *heads* to perform translation, prediction, summarisation, text completion and other linguistic tasks applicable across a wide range of textual contexts.

Important from the point of view of implementation, these attention heads process input data in parallel, i.e. they involve computational processes that act independently until brought together into the relevant layers of a neural network. The substantial volume of computation required to calculate and integrate attention matrices benefits from hardware configurations that support parallel processing, such as

batteries of GPUs (graphics processing units).

AI agents in general

The performance of attention heads in NLP provides a point of contact with other computational methods in urban planning and design, especially if we think of attention heads as computational agents. This similarity provides an opportunity to compare the functioning of conversational AI systems with other computational and AI models developed over many decades and applied in various problem solving domains, not least in engineering, architecture, and design. (For example, a recent search in the cumulative index of publications in architecture and computer-aided design <https://papers.cumincad.org/> reveals over 200 publications with “agent” in the title, dating back to 1990.) That said, such agent-based models have yet to achieve the rapid success and high profile of language-based tools based on the Transformer methodology such as ChatGPT.

The general class of programs researchers identify as “multi-agent systems” consist of sets of programs as procedural algorithms, rules and databases coded with their own tasks, priorities and behaviours. We assume these computational agents operate independently of one another and they are designed to interact with each other in a decentralised manner.

The overall behaviour of such systems *emerges* from the interactions among individual agents. Non-playing avatars (NPAs) and crowd scenes in CGI animation are obvious examples of multi-agent systems. In the case of movie animations, individual agent actions contribute to the overall movement of a large group of avatars in a convincing dance or battle scene. Researchers might use multi-agent systems in simulations involving game-like scenarios where they study interactions and emergent phenomena within complex systems.

In an urban context, such multi-agent systems might be used for traffic management and control. By simulating the behaviour of individual vehicles as individuated agents, these systems can optimise and analyse traffic flows and monitor points of congestion.

Blackboards

Blackboard systems provide a variation on such agent-based methods. Here, each agent contributes different processes or “knowledge sources” that contribute to solving a problem. Each agent writes their contributions on a shared data workspace. The agents take their turn in interacting with the work space.

Blackboard systems were used in early automated natural language processing, where each knowledge source agent focused on different linguistic aspects of a text, such as syntax, semantics, pragmatics, etc., and would post their findings on the virtual blackboard for synthesis into an interpretation or response.

Applications in urban design are more speculative. The idea is that inputs from architects, engineers, city planners, and public feedback are integrated to progress informed decisions about land use, infrastructure development, and environmental impact. Blackboard systems attempt to automate this kind of interdisciplinary collaboration by providing a platform where different expertise and perspectives

can be brought together effectively. I think of virtual whiteboards such as Miro (<https://miro.com>) where human collaborators share a common workspace. But instead of human agents, the Blackboard methodology introduces automated agents.

Comparing hypotheses

A further, related, approach to both NLP and urban problem solving involves the formulation of rival hypotheses for prediction, sentiment analysis, text prediction, etc. Following this model, problems are addressed as if a tree structure, where the trunk is some initial problem statement and the ever-increasing branch structures delineate pathways to alternative hypothetical solutions and sub solutions.

These models follow a generate-and-test process requiring a means of evaluating partial hypotheses and of pruning unlikely paths from a tree of possible outcomes.

In the context of urban planning and architecture, similar strategies can be applied to modelling complicated decision-making processes, where various potential plans or designs are evaluated and iteratively refined to arrive at an optimal solution. This is the early standard computational model of design and problem solving advanced by researchers such as Christopher Alexander, Newell and Simon. Advances on these methods include the use of so-called genetic algorithms that recruit analogies with DNA-type data strings, random mutation and heritability.

Multiple criteria

Multi criteria optimisation involves a set of methods that treat rival or contributing agents as contiguous variables defining solution spaces.

Again, in research departments rather than practice, this is an approach applicable to urban planning and architecture, where decision-making often involves balancing multiple competing criteria. It involves optimizing several objective function simultaneously. These objectives often have different units or scales and can conflict, meaning that improving one objective may worsen another.

Algorithms articulate and define sets of possible solutions (also known as decision spaces) to find the best trade-offs among the different criteria. Each point in this space represents a different combination of the variables under consideration. A solution is Pareto optimal if no other solution is better in at least one criterion without being worse in at least another criterion. The method rarely identifies a single best solution; instead, it offers a set of Pareto optimal solutions. A human decision-makers then chooses the most suitable solution based on undeclared preferences or priorities among different criteria and new criteria.

Considering the conflictual and fluid nature of design objects, aims, and methods, Herbert Simon introduced to the generate-and-test models of agency the concept of satisficing, the production of the best solution all things considered a compromise between agents as it were.

In architectural design, multi-criteria optimisation might involve optimising for factors such as energy efficiency, construction cost, functionality, and comfort, each of which might be in conflict.

We can think of the process of training a neural network, as in NLP systems, as a multi-criteria optimization challenge [ChatGPT then completed the sentence for me]

where the goal is to find a balance between competing objectives such as maximizing model accuracy, minimizing error rates, ensuring generalizability to new data, and often managing the trade-offs between model complexity and computational efficiency. This process involves iteratively adjusting the network's parameters (weights and biases) to optimize its performance across these different criteria, guided by a cost or loss function that quantifies how well the model is performing on the training data. In doing so, the network learns to make predictions or classifications that are as accurate and robust as possible, given the constraints and complexities inherent in the data it is trained on.

Automation and agency

In this post I'm conflating the elements of several decades of research that have sought to apply computational methods, automation and AI techniques to urban design, planning, management, decision making and problem solving. I am thereby demonstrating something of the legacy and controversies that prepare researchers, practitioners and citizens for the reception and resistances to conversational AI (e.g. ChatGPT). This brief survey was triggered by the characterisation of multi head attention in NLP systems viewed as agents.

Bibliography

- Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge,: Harvard University Press.
- Coyne, R., M. Rosenman, et al. (1990). *Knowledge-Based Design Systems*. Reading, Mass.: Addison-Wesley.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Reading, Mass.: Addison Wesley.
- Hayes-, R., B., A. Garvey, et al. (1986). "Application of the BBI blackboard control architecture to arrangement assembly tasks." • *Artificial Intelligence in Engineering* 1(2): pp. 85-94.
- Latour, B. (2005). *Reassembling the Social: An Introduction to Actor-network-theory* Oxford: Oxford University Press.
- Newell, A. and H. Simon (1972). *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.
- Radford, A. D. and J. S. Gero (1988). *Design by Optimization in Architecture, Building and Construction*. New York: Van Nostrand Reinhold.
- Simon, H. (1969). *The Sciences of the Artificial*. Cambridge: MIT Press.

Note

- Featured image is by Dall-e, prompted by: Please generate an aerial view of traffic congestion exacerbated by a traffic accident.

Category

1. Artificial Intelligence

Tags

1. actor network theory
2. generate-and-test
3. optimization
4. problem solving

Date Created

December 23, 2023

Author

rcoyne99

default watermark