NICE ACTIMIZE

Contextual Artificial Intelligence in Autonomous Financial Crime Management

Insights Article

Danny Butvinik
Chief Data Scientist | NICE Actimize

Introduction

Context plays an important role in Autonomous Financial Crime Management. This is especially true for activities such as predicting context changes; explaining unanticipated trends, behavioral patterns and events and helping to handle them; and assisting to focus attention. However, context is a poorly used source of information in our computing environments. As a result, we have an narrow understanding of what context is and how it can be used. The notion of context is important since it can capture many of the interesting aspects of the way we understand the automation processes, such as dependence, priority, partiality, relativity and locality. In Artificial Intelligence, a number of formal or informal definitions of some notion of context have appeared in several areas. However, all these notions of context are very diverse and serve different purposes. The main challenge arises from a different viewpoint on priority of performance requirements. Here, the lack of context use appears more as a mechanism for presenting knowledge rather than for modeling knowledge. The modeling, representation and use of context appears to be the challenge of the coming years, particularly when we now face very complex problems, large knowledge bases and diversity of data sources. To use context effectively, we must understand what context is and how it can be used, and we must have architectural support. An understanding of context will enable application designers to choose what context to use in their applications. An understanding of how context can be used will help architectures determine what context-aware behaviors to support in their solutions. In this article, we provide an overview of context in Artificial Intelligence and its consequences on automation systems.

Our definition of context

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Context in Knowledge Acquisition

Knowledge acquisition is a difficult and time-consuming task. The difficulty arises because experts do not report on how they reach a decision. As a result, the decision is acquired out of its context when experts rather justify why the decision is correct within a specific context. Context in knowledge acquisition is generally considered either before the building of the system or during the use of the system.

Encoding knowledge as part of the task at hand, leads to contextualizing knowledge acquisition. Knowledge is then encoded into the system when it is needed, i.e. in its context of use. Contextualizing the knowledge acquisition process helps ensure that relevant knowledge is put in the knowledge base, because one doesn't know what is really needed until one is in the design process. We are now in the realm of **incremental knowledge acquisition.**

There are different ways to acquire knowledge in context:

- Capture the context by entering the expert's new rule directly as provided, including an 'IF LAST_FIRED (rule n°)' condition. That is, the new rule will not fire on a case unless the old rule that produced the wrong interpretation has fired first. Thus the new rules are tested precisely in the context in which the expert provided them. The portion of the expert system that comes before this rule is exactly the same as the expert system which produced the interpretation. (This representation is used in a 'Knowledge Dictionary'). Consider a justification-based knowledge acquisition to divide the load in knowledge acquisition between a cooperative user/teacher and elicitation program. The machine provides the computational medium, including the knowledge representation and the context of use, so that everything acquired from the user is assimilated into the computational model. The knowledge acquired using the justification technique is guaranteed to be operational, because the user always conveys something to the machine by getting the machine to say it.

- Analyze and represent knowledge of task-driven approach for an intelligent tutoring system (ITS). The approach provides the context to ensure that the knowledge acquired and represented is the knowledge required to support task performance.
- Access to a model of the problem-solving process that will help users select an action. It is acknowledged that the governing model contextualizes the information used.

The main claim of all these approaches is that experts provide their knowledge in a specific context and the knowledge can only be relied upon in this context. The context is largely determined by the case which prompted the change to the knowledge base. Knowledge is not to be generalized when acquired. It is fundamental to record the context in which the knowledge is acquired. However, the acquisition of knowledge in context is still a challenge. Even what we take to be a highly stable behavior, such as reciting a phone number, is highly contextual. You establish this context by sitting in front of a phone. Such a situated knowledge is not acquired with the classical tools prior to its use. Acquiring the knowledge when needed implies that the knowledge is compatible with the computational medium of the system. For instance, one only "sees" the sequence of physical positions that the finger must have on the phone keyboard. It is rather difficult to acquire such a knowledge that may be expressed in a representation formalism that is not known by the machine. Moreover, making explicit context permits one to examine knowledge in any other context. One advantage is that it allows different contexts to be compared, in particular how correcting knowledge in a particular context may be used to review conclusions drawn in another context. Although the approaches described present serious weaknesses, they are ascribed in the realm of the incremental knowledge acquisition in context.

Context in Machine Learning

The meaning of many concepts heavily depends on some implicit context, and changes in that context can cause more or less radical changes in concepts. Incremental concept learning [1] in such domains requires the ability to recognize and adapt to such changes. Widmer [2] presents a general two-level learning model, and its realization in the METAL(B) system. The system learns to detect certain types of contextual clues, and reacts accordingly with context changes. The model consists of a basic learner that performs the regular online learning and classification task, and a meta-learner that identifies potential contextual clues. The operational definition of contextual attributes is based on the notion of predictive features. A feature is considered as a contextual clue if it does not directly determine or influence the class of an object. (Thus, it does not intervene in the learning explicitly.) However, there is a strong correlation between its temporal distribution of values and the time when certain other attributes are predictive. Intuitively, a contextual attribute is one that could be used to predict which attributes are predictive at any point in time. (Thus, contextual clues constrain the learning.) In a similar spirit, Turney [3] reviews five heuristic strategies for handling context-sensitive features in supervised machine learning from examples to recover hidden (implicit, missing) contextual information. Then he presents two methods for recovering two lost (implicit) contextual information.

Park and Wilkins [4] describe a failure driven learning with a context analysis mechanism as a method to constrain explanations and thereby increase the number of learning opportunities by 17% and increases the overall amount of improvement to the expert system by 10%. The context analysis program maps an observed action on the explanation plane. An explanation on this plane has a pointer to a set of actions that are explained. Such an explanation becomes a sub-context that explains a subset of observed actions. The context analysis program can find a sub-context that explains all the actions. This sub-context is considered as the context of the observed actions. Again, the context does not intervene directly in the learning, but constrains it. Here, contexts are judged similar if their strategy axes are the same and their focus axes can be grouped by a known relation. With such approaches, machine learning is considered part of the task at hand and appears very close to incremental knowledge acquisition [5].

Context in Databases and Ontologies

The main role of context here is to provide humans with a much greater control over the knowledge. Context permits users to define which knowledge should be considered, what are its conditions of activation and limits of validity, and when to use it at a given time. This is especially important for the building and the use of large and reliable knowledge systems.

Contexts act like adjustable filters for giving the right meaning in the current context and to present the minimal number of information pieces and essential functions that are necessary to the task at hand. For instance, the concept of water is viewed differently by a thirsty person, the plumber, the chemist, and the painter.

As an alternative to the integration approaches in the literature, Goh et al. [6] propose a strategy based on the notion of context interchange in databases. The Context Interchange strategy is an approach for achieving interoperability among heterogeneous and autonomous data sources and receivers. Context refers to the (implicit) assumptions underlying the way in which an interoperating agent routinely represents or interprets data. Data contexts, as event scripts, are abstraction mechanisms which allow us to cope with the complexities of life. In the context interchange framework, assumptions underlying the interpretations attributed to data are explicitly represented in the form of data contexts with respect to a shared ontology that reduces the cost of communication among members of a group and constitutes a shared vocabulary for context definition. The approach enables the distinguishing of the source (export) and receiver (import) contexts. A context mediator is used to compare the source and receiver contexts and detect any conflicts. The context mediator acts as a contextualization process**.** The export context captures those assumptions integral to the “production” of data in the data source, and the import context captures those assumptions which the data receiver will employ in interpreting the data.

Sciore et al. [7] have proposed an extension to SQL, called Context-SQL (C-SQL), which allows the receivers' import context to be dynamically instantiated in an SQL-like query. Context-SQL provides a vehicle for users who are interested in modifying their import contexts dynamically as queries are formulated.

Walther et al. [8] use the PROTEGE-II system as a meta-tool for constructing task-specific expert-system shells. The system associates each method with an ontology that defines the context of that method. All external interaction between the method and the world during the method assembly are a mapping of knowledge between the method's context ontology and the ontologies of the methods with which it is interacting. Each ontology contains all necessary information for defining the module's role in the module-assembly process, and thus places a number of further requirements on the representation language. This association is described in the context-definition language called MODEL. Shareable ontologies are then a fundamental precondition for reusing knowledge, serving as a means for integrating problem-solving, domain representation, and knowledge-acquisition modules. A shared context is referred to as an ontology because the domain ontology provides a common understanding of the involved design concepts and of the topological relations between them. The context in PROTEGE-II:

1) captures the role of the component during assembly,
2) describes the knowledge required by the component,
3) specifies the input and output requirement of the component, and
4) encapsulates the behavior of the component so it can be reused and shared.

A context is defined as a consistent set of propositional assumptions about which something can be shared. Such a set forms a theory of some topic, e.g., a theory of mechanics, a theory of the weather in winter, etc. In that sense, a context is called a "Micro-Theory." The scope of a context (the theory associated with the context) is the set of objects over which its predictions hold. There are as many contexts as sets of assumptions under consideration. Based on a statement made about an object in one context, something may be derived about that object in another context. The two contexts use different vocabularies and make different attributions of an object, but these attributions are about the same object. So there may be some contexts in which P is not be stateable (in the vocabulary of that context), and there may be other contexts in which P is stated differently. Different expressions might be used by different contexts for stating the same fact. The same expression might also be used by different contexts for stating the same fact or the same expression might mean different things in different contexts. The meaningfulness of a formula may depend on the context in which it occurs. The use of different contexts permits the use of different languages. Huhns et al. [9] use the CYC knowledge base as a context. Then, a model is a set of frames and slots in a CYC context created especially for it. The mapping between each model and the global context (the CYC knowledge base) is captured in a set of articulation axioms. The models of individual resources are compared and merged with CYC, but not with each other, making a global context much easier to construct and maintain. The authors find that using CYC is significant, because of

1) its size: it covers a large portion of the real world and the subject matter of most information resources;
2) its rich set of abstractions: the process of representing predefined groupings of concepts;
3) its knowledge representation mechanisms: a global context can be constructed, represented, and maintained, and
4) its typing mechanism: the integration and checking of the consistency of query results are ensured.

Farquhar et al. [10] present an approach to integrating disparate heterogeneous information sources. They show that the use of context logic reduces the up-front cost of integration path and allows semantic conflicts within a single information source or between information sources to be expressed and resolved. Two contexts are used to represent each information source. The information source context is a direct translation of a database schema into logic without resolving semantic conflicts, so that the translation can be done automatically. The semantic context holds the translation with the semantic conflicts resolved. An integrating context contains axioms that lift sentences from several semantics (or integrating) contexts. The consequences of using context logic to integrate information sources are:

- integrate new information sources incrementally;
- share assumptions among information sources without making them explicit;
- exploit shared ontologies; and
- provide a richer model of integration that goes beyond global schema or federated schema methodologies.

Context in Modelling and Representation

More recently, McCarthy [11] defined a context as a generalization of a collection of assumptions. Contexts are formalized as first class objects (formal objects), and the basic relation is $ist(C, P)$. It asserts that the proposition p is true in the context c, where c is meant to capture all that is not explicit in p, which is required to make p a meaningful statement representing what it's intended to state. Formulas $ist(C, P)$ are always asserted within a context, i.e., something like $ist(c', \ ist(C, P))$: c': $ist(C, P)$. The consequences are:

1) a context is always relative to another context,
2) contexts have an infinite dimension,
3) contexts cannot be described completely,
4) when several contexts occur in a discussion, there is a common context above all of them into which all terms and predicates can be lifted.

The logical machinery is only a small fraction of the effort involved in building a context-based system. The bulk of the effort lies in writing the axioms describing and interrelating contexts. The structure and content of these axioms--the lifting rules--are heavily dependent on the kind of use. The most common operation on contexts is to lift a formula from one context into another. Doing this requires a relative (partial) *de-contextualization*, i.e., the differences between the origin and target contexts had to be taken into account to obtain a formula with the same truth conditions as the original formula had in the origin context. (Note that a formula relating two contexts could involve contextual assumptions and is therefore itself in a context.) The context of the system is the current context of the problem solving. All interactions with the system take place in this context, and information must be lifted from other contexts into this current Problem Solving Context. The current context is the physical/real memory, the other contexts are the virtual memory. Contexts of a problem solving task are usually created dynamically by the system and are ephemeral.

There are many other relations among contexts and context valued functions. For instance, there is a general relation that *specializes* between contexts, specialize(c_1, c_2), that indicates c_2 involves no more assumptions than c_1 and every proposition meaningful in c_1 is translatable into one meaningful in c_2 (inheritance of $ist()$ in both from the sub-context to the super-context and vice versa). Mechanisms for relating and translating between contexts are acknowledged as vital to the effective reuse of domain theories in new problem solvers. There are two classes of context:

1) Representational context that captures the total set of qualifications relative to which the symbols in the language of a theory are abstracted at a pertinent level of relevance; and
2) Computational context that represents the focus of the reasoning--the set of assumptions made or path taken by a reasoner in evaluating a current hypothesis.

Other Representations of Context

There are several theoretical approaches (other than those described before) that consider context – either explicitly or not. Most of these approaches try to represent context (i.e. account for context), not model it. Note that we distinguish model and representation. The goal of a model is to give a coherent picture of context that can be used for explaining and predicting by simulation. The goal of a representation of context is only to account for what is observed whatever that may be. A model is endowed in a theory, and a representation lies on the representation formalism chosen. As the two goals are different, we consider here modeling at the theoretical level and representation at the programming level. A successful modeling is a modeling that is used in applications.

Context has been considered on the basis of the Situation Theory [12]. Situation theory is a unified mathematical theory of meaning and information content that is applied to specific areas of language, computation and cognition. The theory provides a system of abstract objects that make it possible to describe the meaning of both expressions and mental states in terms of the information they carry about the external world.

Ezhkova [13] defines context in knowledge representation techniques on the basis of the concept of contextual system (CS). The main purpose of a CS is stated from the viewpoint of the decision-making problem. According to this viewpoint, a global problem is reduced to a set of local problems where each one is stated for a specific set of decision alternatives and requires formation of the context for its own problem area. A contextual system has two types of memory: a *long-term memory* (a primary database and a base of contexts) and a *short-term memory* (intra-context knowledge processing and inter-context knowledge processing). An algebra of contexts is proposed to involve contraction, extension, immersion, coupling and intersection of contexts. Contexts are then stored or dynamically generated. For example, by contracting a context, one may focus on certain sections of its description. There are different types of context contraction with respect to attribute significance, a set of attributes, the number of the most significant attributes, and a set of basic concepts. Conversely, the context expansion operations are required for further learning, adaptation, introduction of new concepts, and immersion in a larger context. Behind this concept is the problem area context (PA), which is a meta-notion relative to knowledge bases. The PA permits to determine distances between the concepts of the context, the proximity of concepts behind decision-making schemes being largely dependent on the problem area context. Two concepts may be close in one context and diametrically opposite in another one. The introduction of the contextual space makes it possible to determine a distance between the concepts of the context. The distance is described in a universal manner. The distance between concepts in the context space is behind the intra- and inter-context processing.

The inter-context knowledge processing employs context algebra and logic. The former supports processes, such as context contraction and extension, submersion of the given context in a wider one, and integration and intersection of contexts, which offer an interesting interpretation and formalize different non-traditional knowledge processing schemes.

Summary and Discussion

Our interest for context is for the automation in the financial crime management domain. In this article, we studied more specifically the link between context, explanation and incremental knowledge acquisition. We saw that context plays an important role in financial domain where activities imply reasoning and interpretation, and can only be caught by experience. This interest for the use of context implies that there is no clear and general definition of context. Context seems to possess, according to the domain, a double nature: static or dynamic, discrete or continuous, knowledge or process. This apparent double nature arises from the fact that the notion of context is dependent in its interpretation on a cognitive science versus an engineering point of view. This explains why there is a theory-versus-practice gap, and why it seems difficult to attempt to unify the various notions of context as long as a consensus is not reached. As a consequence, one considers context as a concept with complex topology, an ontology, a shared space of knowledge, a consistent set of propositional assumptions, a semantic background, the environment of communication, a set of restrictions that limit the access to parts of a system, etc.

The context is considered as something that is stored in long-term memory, and recalled as a whole, as a viable unit of task strategy appropriate to some stage of some task. Moreover, a contextual system may have two types of memory: a long-term memory (a primary database and a base of contexts) and a short-term memory (intra-context knowledge processing and inter-context knowledge processing). A context may also be generated dynamically and, according to McCarthy [11], created from old contexts. The difficulty here is to determine if one needs to store all past contexts or have a set of elementary contexts that may be combine to constitute complex contexts to adequately represent a particular situation. McCarthy [11] points out that the logical machinery is only a small fraction of the effort involved in building a context-based system. The bulk of the effort lies in writing the axioms describing and interrelating contexts. A solution for ensuring a correct transfer of information from one context to another may be a context manager. A context manager makes compatible the interpretations (or reasonings) in the source and destination contexts. This is indeed more a process of contextualization than a part of the knowledge. The context manager is supposed to:

- retain as much of the knowledge generated as possible
- provide easy access to and a good explanation of this knowledge
- make the best use of the knowledge already held in the dynamic knowledge base, enabling it to generate new knowledge without performing redundant inference
- help the user compare different, sometimes conflicting solutions

For example, a good context manager would make compatible users' requests and the conceptual schema of a database. However, a context manager acts at the level of the presentation of the knowledge rather than its representation (or modelling). Therefore one often gives context the role of filter at the programming level. The double action of a context manager on knowledge at a given step of a problem solving is:

(i) selecting the knowledge pieces for the focus of attention
(ii) keeping in stand-by other knowledge pieces

However, it stays to make the different contexts compatible. Contexts define when the knowledge should be considered and thus simplifies the construction of the knowledge base by imposing requirements on the representation language. These requirements structure knowledge bases in tractable units, often organized in a hierarchy. A context contains:

(i) sets of concepts (also called schemas, frames, or structures) that describe the basic terms used to encode knowledge in the ontology, and
(ii) a set of constraints that restrict the manner in which instances of these concepts may be created and combined. Thus, context-encapsulated knowledge appears as a chunk of reasoning.

A challenge here is how knowledge in its context of use may be examined in other contexts (a kind of knowledge de-contextualization). However, the relationships between context and knowledge are yet to explored. A piece of knowledge may be contextual or contextualized according to where we are in terms of problem solving. Contextualized knowledge is knowledge that is explicitly considered in the problem solving. Contextual knowledge intervenes implicitly in the problem solving, most often as constraints. At the following step of the problem solving, contextualized knowledge may become contextual knowledge at the new step or external to the step. An operational definition of context requires to take into account if we're speaking of problem solving as a whole or of a given step of the problem solving.

In the formula $ist(C, P)$, McCarthy [11] defines context to capture all things that are not explicit in P but that are required to make P a meaningful statement representing what it's intended to state. In other words, a context is a structure, a frame of reference, that permits not saying all the things in a story. Context permits implicit things that do not intervene directly in the problem solving. A first step towards an operational definition of context is to define context as what constrains problem solving without intervening in it explicitly.

In contextual artificial intelligence, the efforts concern mainly the representation of context nested with the knowledge. Some of the questions that must be addressed are:

1. Does part of the context belong to the knowledge base or a particular context base?
2. What are the relationships between context and meta-knowledge, context and knowledge representation, context and time, context and decision?
3. What are the relationships between contextualization process and control knowledge?

As a result, the concept of contextual artificial intelligence remains open for discussion.

References

[1]Incremental learning: https://en.wikipedia.org/wiki/Incremental_learning

[2] Widmer, G.: Recognition and exploitation of contextual clues via incremental meta-learning (Extended version). Oesterreichisches Forschungsinstitut fuer Artificial Intelligence, Wien, TR-96-01, 1996.

[3] Turney, P.: The management of context-sensitive features: A review of strategies. Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains, 1996, pp. 53-69.

[4] Park, Y.-T.—Wilkins, D.C.: Establishing the coherence of an explanation to improve refinement of an incomplete knowledge base. Proceedings of the AAAI90 1990.

[5] Damminda A.: Incremental knowledge acquisition and self-learning from text. Conference paper DOI: 10.1109/IJCNN.2010.5596612

[6] Goh, C.H.—Madnick, S.E.—Siegel, M.D.: Context interchange: Overcoming the challenges of large-scale interoperable database systems in a dynamic environment. 1995, In Proceedings of the Third International Conference on Information and Knowledge Management, Gaithersburg (USA) 1994, pp. 337--346.

[7] Sciore, E.—Siegel, M.—Rosenthal, A.: Context interchange using meta-attributes. Proceedings of the 1st International Conference on Information and Knowledge Management, 1992, pp. 377-386.

[8] Walther, E.—Eriksson, H.—Musen, M.A.: Plug-and play: Construction of task-specific expert-system shells using sharable context ontologies. Proceedings of the AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition, San Jose, CA, pp. 191-198. AAAI, 1992. Technical Report KSL-92-40, Stanford (USA) 1992.

[9] Huhns, M.N.—Jacobs, N.—Ksiezyk, T.—Shen, W.-M.—Singh, M. P.—Cannata, P.: Enterprise information modeling and model integration in Carnot. Proceedings of the International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Netherlands, May 1993, pp. 12-14.

[10] Farquhar, A.—Dappert, J.—Fikes, R.—Pratt, W.: Integrating information sources using context logic. KSL-95-12, Knowledge Systems Laboratory (USA), January 1995.

[11] McCarthy, J.: Notes on formalizing context. Proceedings of the 13th IJCAI, Vol. 1, 1993, pp. 555-560.

[12] Situation Theory: https://en.wikipedia.org/wiki/Situation_theory

[13] Ezhkova, I.V.: Contextual systems: Is it a way of a universal expert system development?. In: General systems, G. Klir Publisher, New Jersey, USA 1992.

Learn more about NICE Actimize Financial Crime Analytics research here.

ABOUT NICE ACTIMIZE

NICE Actimize is the largest and broadest provider of financial crime, risk and compliance solutions for regional and global financial institutions, as well as government regulators. Consistently ranked as number one in the space, NICE Actimize experts apply innovative technology to protect institutions and safeguard consumers and investors assets by identifying financial crime, preventing fraud and providing regulatory compliance. The company provides real-time, cross-channel fraud prevention, anti-money laundering detection, and trading surveillance solutions that address such concerns as payment fraud, cybercrime, sanctions monitoring, market abuse, customer due diligence and insider trading.

info@niceactimize.com | www.niceactimize.com/blog | @nice_actimize | /company/actimize | NICEactimize

NICE · ACTIMIZE