Shifting and drifting attention while reading: A case study of nonlinear-dynamical attention allocation in the OpenCog cognitive architecture

Article *in* Biologically Inspired Cognitive Architectures · July 2018

DOI: 10.1016/j.bica.2018.07.005

4 authors, including:


Matthew Ikle
SingularityNET


David Hanson
Hanson Robotics

Shifting and Drifting Attention While Reading: A Case Study of Nonlinear-Dynamical Attention Allocation in the OpenCog Cognitive Architecture

Misgana Bayetta Belachew[1,2,3] and Ben Goertzel[1,2,3] and Matthew Ikle' [4] and David Hanson[2 3]

[1] OpenCog Foundation, [2] SingularityNET, [3] Hanson Robotics, [4] Adams State University

Abstract. A simple experimental example of the general principle of "cognitive synergy" underlying the OpenCog AGI architecture is explored: An OpenCog system processing a series of articles that shifts from one topic (insects) to another (poisons), and using its nonlinear attention-allocation dynamics (based on the ECAN Economic Attention Networks framework) to spread attention back and forth between the nodes and links within OpenCog's Atomspace knowledge store representing the words in the sentences , and other nodes and links containing related knowledge.

With this setup, we study how the ECAN system shifts the attentional focus of the system based on changes in topic – in terms of both the speed of attention switching, and the contextual similarity of the content of attentional focus to the sentences being processed at a given point in time. This also provides an avenue for exploring the effects of particular design choices within the ECAN system.

For instance, we find that in this particular example, if the parameters are set appropriately, ECAN indeed causes the system to assign particular importance to nodes and links related to the "insecticide" concept, when it is reading sentences about poisons in a situation where it has been primed by sentences about insects. This is an example of what we call "drifting" attention – the system's attention moves to something suggested by its perceptions, even if not directly presented in them. .

1 Introduction

One approach to creating AGI systems is the "integrative" strategy, involving combining multiple components embodying different structures or algorithms, and relying on synergistic dynamics between components. One kind of integrative system involves various highly independent software components, each solving a specialized set of problems in a mostly standalone manner, with occasional communication between each other in order to exchange problems and solutions. On the other end of the scale, are systems designed as tightly interconnected components that give rise to complex non-linear dynamical phenomena.

Here, we are specifically focused on the latter approach. We will discuss the particulars of one form of cognitive synergy – between natural language processing agents and nonlinear-dynamical attention allocation – within the context of one particular integrative AGI architecture, the OpenCog platform [4] [7] [8] (and using OpenCog according to a cognitive architecture now referred to as PrimeAGI, previously referred to as OpenCogPrime).

The core cognitive processes involved here are:

- Natural language processing that reads a sentence and creates a number of nodes and links (Atoms) representing syntactic and semantic relationships between the words in the sentence (represented as WordNodes in the OpenCog Atomspace hypergraph knowledge store) and their associated concepts and relationships
- Attention allocation that manages the spreading of STI (Short Term Importance) values among Atoms in the Atomspace. Atoms of initial interest are stimulated with STI, and then Atoms spread STI to other Atoms that they are linked to, according to the nonlinear-dynamical equations of the ECAN module.
- The Attentional Focus, defined as the set of top-STI Atoms in the Atomspace, is updated via slightly different dynamics than the rest of the Atomspace; STI values of Atoms in the Attentional Focus are updated more frequently and consistently, and there is an optional process that builds HebbianLinks between pairs of Atoms in the Attentional Focus

The specific dynamics explored and demonstrated here, using these cognitive processes, is as follows:

- The OpenCog NLP Pipeline processes (doing syntax parsing and then some light semantic interpretation) a series of articles – first insect related articles and then poison related articles
- The WordNodes of each sentence parse created in the OpenCog Atomspace during this NLP processing are provided a “stimulus boost” , thus giving them a high ”Short Term Importance” (STI) value and triggering (ECAN) attention dynamics
- Since insecticides are related to insects, during the reading of insect-related articles, Atoms related to insecticides will get enough STI via importance-spreading to be at the fringe of the Attentional Focus (and some may also end up in the Attentional Focus as well, though for a relatively short period of time compared to other more centrally insect related WordNodes).
- During the parsing of the poison articles, once again insecticides get STI boost via the effect of STI spreading; this causes more of them accumulate enough STI to enter into the attentional focus and stay for a relatively longer duration compared to previously.

This is exemplified in Figure 3 in Section 5 below. This is clearly a cognitively meaningful behavior, which exemplifies a type of dynamics that is crucial for efficient reasoning and processing of perceptual information where there is constant switching of topics. Among other things, this is an example of what we call "drifting" attention – the system's attention moves to something suggested by its perceptions, even if not directly presented in them. This occurs together with "shifting" attention, and the two intertwine in complex ways.

This is a preliminary study and the phenomena noted need to be studied in a much more thorough and rigorous way in order to achieve definitive results or create a valid basis of comparison with other cognitive architectures or human data. However, many of the observations made in this study appear cognitively sensible and worthy of further investigation. For instance, a prior study [10] analyzed some phenomena arising upon utilizing ECAN attention allocation together with PLN (Probabilistic Logic Networks) inference. A natural next step will to put these two aspects together, and utilize ECAN, natural language processing, and PLN reasoning (on logical knowledge extracted from natural language) all together.

2 Background: The OpenCog Framework

Our work here is based upon specific details of the OpenCog AGI architecture, which are too voluminous to review in detail here; we will give some brief sketches and then refer the reader to the literature.

2.1 Memory Types and Cognitive Processes in OpenCog

OpenCog's memory types are the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience [14], plus attentional memory for allocating system resources generically, and intentional memory for allocating system resources in a goal-directed way. Table 1 overviews these memory types, giving key references and indicating the corresponding cognitive processes, and which of the generic patternist cognitive dynamics each cognitive process corresponds to (pattern creation, association, etc.).

The essence of the PrimeAGI cognitive architecture, being implemented and experimented with using OpenCog, lies in the way the structures and processes associated with each type of memory are designed to work together in a closely coupled way, the operative hypothesis being that this will yield cooperative general intelligence going beyond what could be achieved by an architecture merely containing the same structures and processes in separate "black boxes."

2.2 OpenCog's Natural Language Processing pipeline

Opencog's NLP pipeline, in its most commonly used form, consists of three main components: The CMU Link Grammar natural language parser, Relex and Relex2Logic.

RelEx, a narrow-AI component of OpenCog, is an English-language semantic dependency relationship extractor, built on the Carnegie-Mellon Link Grammar

Memory Type	Specific Cognitive Processes	General Cognitive Functions
Declarative	Probabilistic Logic Networks (PLN) [3]; concept blending [2]	pattern creation
Procedural	MOSES (a novel probabilistic evolutionary program learning algorithm) [12]	pattern creation
Episodic	internal simulation engine [6]	association, pattern creation
Attentional	Economic Attention Networks (ECAN) [9]	association, credit assignment
Intentional	probabilistic goal hierarchy refined by PLN and ECAN, structured according to MicroPsi [1]	credit assignment, pattern creation
Sensory	In current OpenCog R&D, this is supplied by semantic deep neural nets, together with other functions such as reasoning, attention allocation, pattern creation, credit assignment	

Table 1: Memory Types and Cognitive Processes in OpenCog Prime. The third column indicates the general cognitive function that each specific cognitive process carries out, according to the patternist theory of cognition.

parser. It uses a series of hand-coded graph rewriting rules to identify subject, object, indirect object and many other syntactic dependency relationships between words in a sentence. That is, it generates the dependency trees of a dependency grammar. The set of dependency relations it employs are quite similar to those of Dekang Lin's MiniPar, the Stanford parser and Google's SyntaxNet. (It even has an explicit compatibility mode with the Stanford parser). It is inspired in part by the ideas of Hudson's Word Grammar.

RelEx2Logic (R2L) produces a certain form of predicate-argument structure for an English-language sentence, transforming the output of RelEx into a logic-based structure using a set of hand-coded transformation rules. The structure produced by R2L roughly resembles classical predicate logic, with aspects of term logic as well.

Sentence parsing in OpenCog, as a practical software process, utilizes all the above three tools in a sequential manner – starting with feeding raw sentence text to the RelEx server which extracts relations between words of the sentence using link grammar – and passes it to Relex2Logic for conversion to OpenCog's logical format, which results in logical Atoms being stored in the Atomspace.

The Next Generation OpenCog NLP Pipeline The specific components of this pipeline are likely to be replaced in the next 1-2 years via corresponding components founded on learning rather than hand-coded rules. Current research on unsupervised grammar induction using OpenCog, aims to replace the hand-coded grammatical lexicon of the link parser via a similarly-formatted grammar whose rules are learned from unlabeled corpus analysis [15]. Preliminary results from this approach are promising [11]. Current research extracting transformation rules from a parallel English/Lojban corpus aims to replace the RelEx and R2L rulebase with similar (but more elegantly structured) rules learned by supervised learning and powered by the Lojban ontology [5] [16].

These advances will improve the flexibility and coverage of the pipeline via grounding it more thoroughly and learning, and thus will bring the framework closer to AGI; however, the dynamics of the ECAN/NLP interactions explored in this paper should not be dramatically affected by these changes. This fact provides a partial validation of the development strategy of using hand-coded rules as scaffolding along the way to creating a fully learning-based system. The hand-coded rules are too crude to yield human-level linguistic capability, but they do provide a holistic working system that can be used to explore other phenomena such as the intersection of language and attention, in parallel with the learning-based NLP pipeline being built.

3 Economic Attention Networks

The Economic Attention Network (ECAN) system, which handles attention allocation within OpenCog, can be viewed as a subhypergraph of OpenCog's general Atomspace hypergraph, consisting of various untyped nodes and links that are used for other purposes but also have attention values associated with them, and also of special attention-management links that may be typed either HebbianLink or InverseHebbianLink. Each Atom in an ECAN is weighted with two numbers, called STI (short-term importance) and LTI (long-term importance); while each Hebbian or InverseHebbian link is weighted with a probability value. A system of equations, based upon an economic metaphor that envisions STI and LTI values as artificial currencies, governs importance value updating. These equations serve to spread importance to and from various Atoms within the system, based upon the importance of their roles in performing actions related to the system's goals.

An important concept in ECAN is the Attentional Focus, consisting of those Atoms deemed most important for the system to achieve its goals at a particular moment. In the context of a dialogue system, for instance, ECAN's key role would be attracting similar concepts and predicates into the Attentional Focus via importance spreading when a particular word from a sentence read or heard by the system is stimulated. This enable NLP and cognition (e.g. PLN) agents to come up with conclusions which can then be directly applied to constructing a reply sentence.

3.1 Scaling Up and Improving ECAN

Due to scalability issue of the previous ECAN implementation, we have changed the ECAN software architecture within OpenCog dramatically during the last year. These changes are documented in depth in markdown files in the associated code repository, `https://github.com/opencog/opencog/tree/master/opencog/attention`. These changes were mainly inspired by the need to apply ECAN in real-world OpenCog usage scenarios, including the control of the Sophia robot created by Hanson Robotics, which is now being controlled by OpenCog and the Hanson AI framework on an experimental basis. In these scenarios, there are multiple concurrent agents accessing the Atoms in the Atomspace in a real-time way, requiring a highly efficient and stable ECAN dynamics.

The attention dynamics algorithms for the Attentional Focus and the rest of the Atomspace have now been separated – where the attentional focus uses an iterative algorithm similar to that from previous implementation, and the one working on the rest of the Atomspace has been modified to use stochastic selection. This change has drastically improved the speed performance of the ECAN system.

Another important change was the redefinition of the concept of Attentional Focus. Instead of using a threshold STI value as a boundary to distinguish the Attentional Focus from the rest of the Atomspace, we now use a fixed size set holding the top K STI valued atoms as the Attentional Focus, where K is an adjustable parameter. This solved some stability problems observed in the previous implementation.

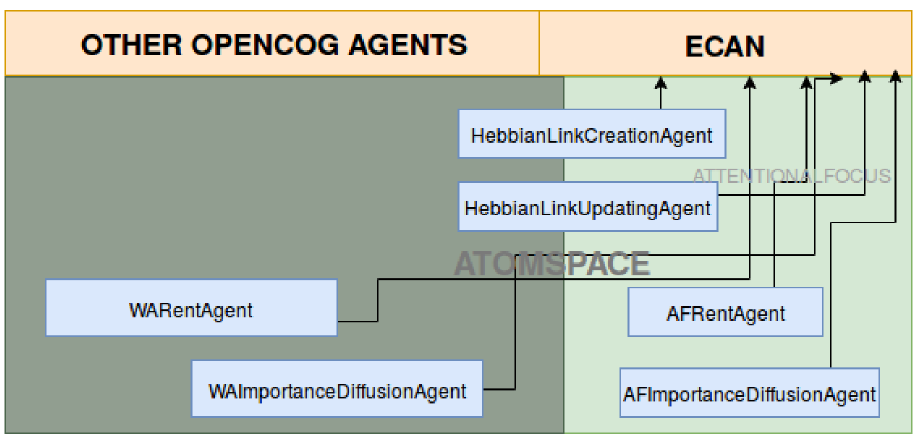


Fig. 1: The Figure shows the key components of ECAN and how they access the Atomspace (the two distinctive colored region of the lower rectangular region depict the Attentional Focus and the rest of the Atomspace). AFRentAgent and AFImportanceDiffusionAgent are agents calculating rent and diffusion importance respectively, iterating through all the Atoms in the AF. HebbianLinkCreationAgent and HebbianLinkUpdatingAgent agents are agents responsible for creation and strength update of HebbianLinks respectively. These agents access all Atoms in the Atomspace. WARentAgent and WAImprtanceDiffusionAgent Stochastically select and calculate rent and diffusion over the Atoms in the Atomspace excluding those in the Attentional Focus.

4 Case Study: NLP pipeline + ECAN

We now describe a particular case study that has been a subject of our recent experimentation, regarding interaction between OpenCog's NLP pipeline and its ECAN attention-allocation module.

In this experiment we first fed the OpenCog Atomspace with prior knowledge about relations between English words, drawn from:

- The Wordnet and Conceptnet4 databases
- SimilarityLinks between words, calculated using the Adagram neural network [13] for calculating the probabilistic weights of the SimilarityLinks.

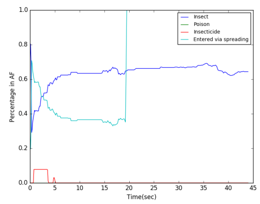
After feeding the system this prior knowledge, we started the ECAN system and also the NLP pipeline, and began to load into the Atomspace a series of articles, beginning with articles regarding insects and then shifting to articles regarding poisons. As the system ingests each sentence in an article, the WordNodes corresponding to the word in the sentence are stimulated with STI from the ECAN module, thus triggering attentional focus dynamics correlated with the reading process.

Amongst the common concepts related to both insects and poison are insecticides. In our experiment, we have curated a list of insecticide names to look for as the attention dynamics changes. What we are curious about here is whether the insecticide related Atoms will pop up and stay in the attentional focus, during the course of the reading process, even when neither the insect nor the poison articles are directly related to them. Thus, we are testing the importance diffusion and dynamics of attention amongst similar Atoms in the atomspace. Another aspect we are observing is the duration of stay of each Atom in the attentional focus – with particular focus here on insecticide related Atoms.

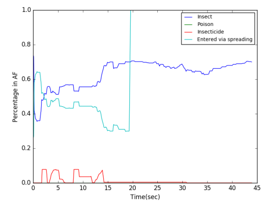
5 Experimental Results

For our early practical exploration with the above setup as reported here, we experimented manually with tuning various of the ECAN subsystem's parameters. The meanings of the parameters varied are described in depth in [8]; the value-sets explored are reported in Table **??**. Some key parameters were not varied across these experiments, for sake of simplicity: The STI/LTI funds buffer, starting STI/LTI funds and target STI/LTI funds were all kept to a constant amount of 100,000, and the "max STI rent value" was kept at a constant value of 1. Figure 2 shows the time-courses of the quantity "duration in Attentional Focus" for Atoms related to insects, poison and insecticide respectively, for each parameter value set explored. The same series of documents is read by the system for each parameter value set.

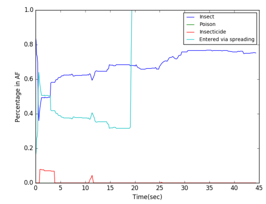
Figure 3 gives a larger view of the results of "Setting 15" , one of several example parameter value sets that successfully displays the key phenomenon being searched for: meaningful spreading of attention to insecticide related Atoms



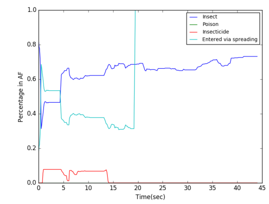
(a) Setting1



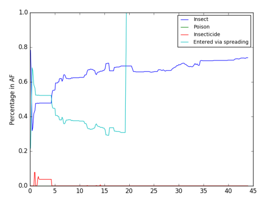
(b) Setting2



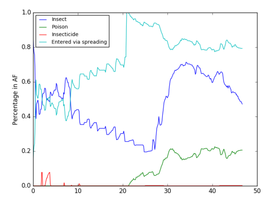
(c) Setting3



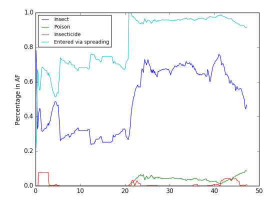
(d) Setting4



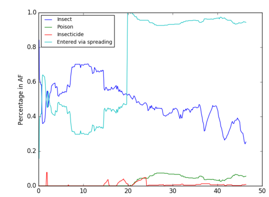
(e) Setting5



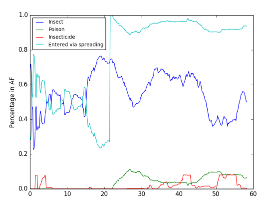
(f) Setting6



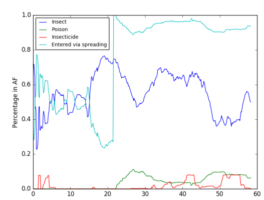
(g) Setting7



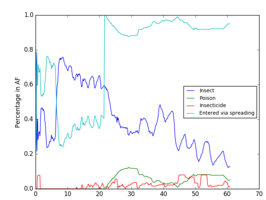
(h) Setting8



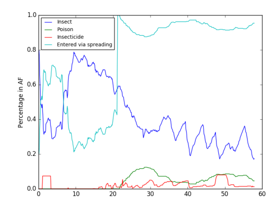
(i) Setting9



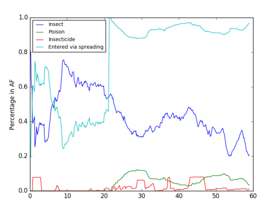
(j) Setting10



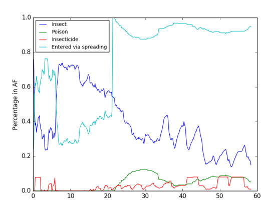
(k) Setting11



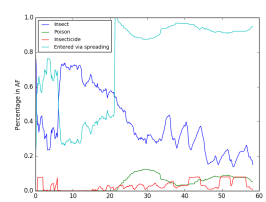
(l) Setting12



(m) Setting13



(n) Setting14



(o) Setting15

Fig. 2: Plots based on Table 2

Table 2: ECAN parameter settings explored in our informal experimentation yielding to the results presented.

	Settings														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
stimulation	500	500	400	300	300	300	50	200	50	50	50	50	100	100	50
rent_frequency	0.001	0.001	0.01	0.01	2	1	0.5	0.5	3	3	5	5	5	5	5
max_spreading_percentage	0.8	0.8	0.8	0.8	0.95	0.95	0.95	0.8	0.95	0.95	0.95	0.95	095	0.95	0.95
max_hebbian_allocation_percentage	0.3	0.3	0.3	0.3	0.3	0.6	0.6	0.8	0.6	0.8	0.6	0.8	0.8	0.9	0.9
Max_AF_size	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500

when the contents of the system's reading includes both poison based and insect based Atoms.

Conceived more broadly, what we see in this relatively simple example is both *shifting* and *drifting* of attention. Shifting attention to related concepts in accordance with shifts in the contents of perception; and drifting of attention to concepts that are associated with the contents of perception, even if not directly contained in the contents of perception. The drifting of attention to insecticide based on ingestion of content related to insects and poison, is an interesting example where attentional dynamics begins to verge on simple cognition.

6 Conclusion

The results obtained here are preliminary and simplistic and need to be followed up with more careful and systematic studies on a greater variety of data, with a greater variety of modes of data analysis and presentation. However, we consider them promising in terms of illustrating the types of phenomena obtained by combining ECAN with perception processing in a system with relevant background knowledge.

Within the scope of ECAN R&D, the next major step is to fully automate the parameter tuning process. For real-world uses of ECAN, the composition of the Atomspace can be expected to be highly nonstationary in multiple respects, so that having a human expert perform parameter tuning in advance, as was done for these examples, cannot work. What must be done is to identify automatically-computable indicators of effective ECAN system function, and identify a set of Atomspace variables (features) on which effective ECAN system function is expected to display significant dependency, and then use (e.g.) reinforcement learning methods to learn functions guiding parameter value choice based on feature values. The specific example explored here is well suited to use as a test case for experimenting with this sort of automated parmeter tuning.

For ECAN in the general context of the OpenCog cognitive architecture and its embedding in the SingularityNET and Hanson AI platforms, a major next step will be studying how this sort of dynamic combines with the dynamics explored in [10] in which ECAN is used to help guide probabilistic logical reasoning. Studies of this nature, exploring the combination of different cognitive processes,

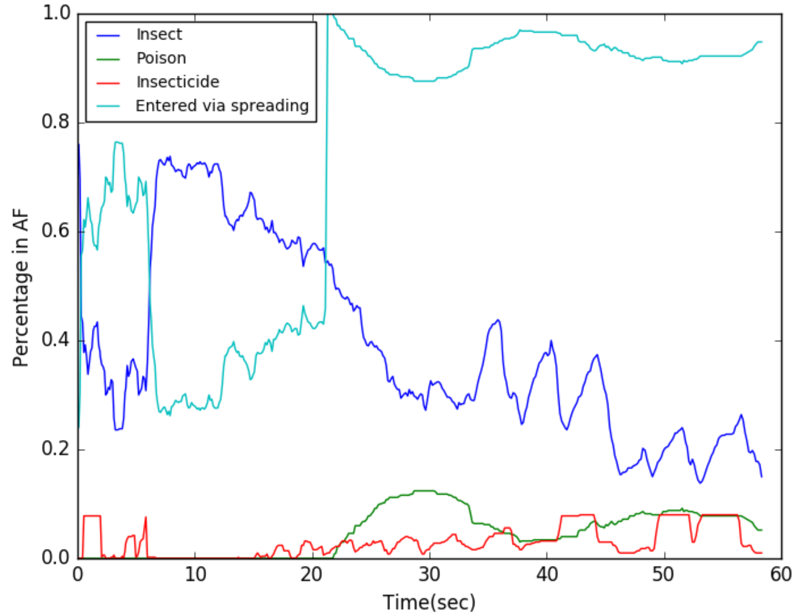


Fig. 3: Shows the duration of stay of insecticide atoms and links containing them in the attentional focus. The curves corresponding to “insect” and “poison” are driven largely by the topics of the articles being read by the OpenCog system at each point in time; the curve corresponding to “insecticide” is driven mostly by indirect ECAN effects, i.e. spreading of STI to insecticide-related Atoms from insect-related and poison-related Atoms, along various links including those created by ECAN.

are key to the larger goal of making all the different components of a human-like cognitive system work effectively together, which is central to achieving powerful AGI if the cognitive-synergy theory underlying the OpenCog architecture is correct.

References

1. Bach, J.: Principles of Synthetic Intelligence. Oxford University Press (2009)
2. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic (2002)
3. Goertzel, B., Ikle, M., Goertzel, I., Heljakka, A.: Probabilistic Logic Networks. Springer (2008)
4. Goertzel, B.: Cognitive synergy: A universal principle of feasible general intelligence? In: Proceedings of ICCI 2009, Hong Kong (2009)
5. Goertzel, B.: Lojban is cool. Some Conference Somewhere (2015)

6. Goertzel, B., Et Al, C.P.: An integrative methodology for teaching embodied non-linguistic agents, applied to virtual animals in second life. In: Proc.of the First Conf. on AGI. IOS Press (2008)
7. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy. Springer: Atlantis Thinking Machines (2013)
8. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 2: The CogPrime Architecture for Integrative, Embodied AGI. Springer: Atlantis Thinking Machines (2013)
9. Goertzel, B., Pitt, J., Ikle, M., Pennachin, C., Liu, R.: Glocal memory: a design principle for artificial brains and minds. Neurocomputing (Apr 2010)
10. Harrigan, C., Goertzel, B., Ikle', M., Belayneh, A., Yu, G.: Guiding probabilistic logical inference with nonlinear dynamical attention allocation. In: Proceedings of AGI-14 (2014)
11. Kolonin, A., friends: Unsupervised grammar induction. submitted for publcation (2018)
12. Looks, M.: Competent Program Evolution. PhD Thesis, Computer Science Department, Washington University (2006)
13. person: Adagram. somewhere (2016)
14. Tulving, E., Craik, R.: The Oxford Handbook of Memory. Oxford U. Press (2005)
15. Vepstas, L., Goertzel, B.: Learning language from a large (unannotated) corpus. CoRR abs/1401.3372 (2014), http://arxiv.org/abs/1401.3372
16. Wiki, O.: From link parse to logic via lojban (2016), http://wiki.opencog.org/xxx/