Exploring the Exocortex: An Approach to Optimizing Human Productivity

Michael Katsevman
Michtom School of Computer Science
Brandeis University
Waltham, MA
mikek@brandeis.edu

***Abstract*—This paper describes the background, design, and prototype implementation of a system for the methodical augmentation of human intellect through interactive means.**

I. Introduction

Originally conceived as an attempt to use biologically inspired machine learning techniques such as neural nets and genetic algorithms toward modeling and then improving day-to-day human behavior, this project moved toward a more direct path to solving that problem.

A. What is the exocortex?

This is a term coined by researcher Ben Houston[1]—and popularized by science fiction author Charlie Stross—to describe the various systems humans may use in thinking but which are not part of our bio-brain. Already, our Blackberries, iPhones, and other essential electronic devices constitute a proto-exocortex.

B. Why work on the exocortex?

As human civilization has grown, it has increased in complexity. Some believe in a future Singularity[2]. The Flynn Effect most likely is a result of humans attempting to adapt to this environment which is growing exponentially more complex[1]. Already the problems of an Attention Economy—a concept pioneered by Herbert Simon[2], who also pioneered modeling human behavior and augmenting human cognition—are apparent: There are more things one must pay attention to, within the same time constraints and physical limitations.

Thus, to cope with this information, and more importantly, attention load humans must create appropriate tools. The exocortex is a collective name for those tools.

II. Previous Work

An early systematic approach to machine involvement with human intelligence is a seminal work by one of the luminaries of human-computer interaction Douglas Engelbart—best known for inventing the computer mouse. In [3] he presents an approach to augmenting human intellectual capabilities.

Engelbart follows the common model of human cognition as a sensory-mental-motor complex. Inputs are provided by the senses, processed via some mental system, and then various motor functions output the results back into the world.

Problems are approached by humans by creating solutions that are broken down into many processes and subprocesses. He calls these process collections *process hierarchies*.

Different process capabilities of an individual—i.e. the actions the individual may perform—form that individual's *repertoire hierarchy*.

In order to figure out how one may augment a human further, one must understand better how we have been augmenting ourselves up to now. So, Engelbart created an experiment that demonstrates ”de-augmenting” an individual. First, the subject wrote *”Augmentation is fundamentally a matter of organization”* using a typewriter, taking only a few seconds. Then, the subject produced the statement in cursive, with a much slower time. Then the next stage of the experiment *”de-augmenting a human by attaching a brick to a pen”* continued by attaching a brick to the pen, writing in cursive; performance time, as well as quality of product, was reduced markedly.

Although the nature of the product itself had no changed much, the efficiency as well as convenience of the activity was greatly reduced. First the elimination of augmenting tools, and then actively reducing the capability of remaining tools resulted in the decreased effectiveness of the human subject.

Augmenting capabilities does not hinge on a particular mental theory, since it is only the selection and efficiency of capabilities that is affected. The exact nature and process of the capabilities is of secondary importance. That is, I am taking an empirical approach to augmentation rather than one that somehow interacts or is dependent on underlying human processes.

Engelbart refers to Vannevar Bushs seminal 1945 article ”As We May Think”[4] where Bush describes a system called a Memex. It essentially amounted to microfilm readers and writers linked over the telephone network. This became a major inspiration for hypertext in general and the World Wide Web specifically. Engelbart goes on on to note that the Memex has but an added benefit of speed and convenience over a traditional filing system. That is, no new capabilities were truly added. Only that instead of walking through a hall of filing cabinets, recall is fast. Much like a phone call is a mere spatial surrogate of talking in person.

One of the reasons that Bushs predictions—perhaps self-

[1] http://www.exocortex.org/ben/
[2] http://en.wikipedia.org/wiki/Technological_singularity

fulfilling since many inventors and developers were inspired by this article—are so apt is that little technological development remains that is not just an externalization of faculties (i.e. capabilities) that were previously performed less efficiently or maybe wholly unaugmented.

Herbert Simon and Allen Newell studied the way in which humans solve problems and created the General Problem Solver[5], a system that when fed with objects and operations would derive solutions. The system had limitations in that to solve any complicated real-world problem the input size and subsequent running time was prohibitive.

Later, as part of the study human problem solving Simon would write about Verbal Protocol Analysis[6] a technique for eliciting stories about activities from people, a variation of which I use.

The other big influence on my approach was Marvin Minsky and his Society of Mind[7]. The major component that affected my thinking was the breaking down of complex cognitive processes into components, eventually rooted in the most basic sensory and motor capabilities, much as Engelbart proposed. Nevertheless, this doesn't imply a fully reductionist system. Higher level processes—what Minsky calls *agents*—are composed of lower level ones but carry some gestalt property.

III. Design

Engelbart lays the foundations of my approach to helping humans achieve goals. I want to derive *process hierarchies* and *repertoire hierarchies* by annotating *strategy narratives* so that the system may select an optimal *process hierarchy* for each goal at each point in time; later the optimal strategy may change based on further input.

A. Conceptual Lexicon

Each human activity I call an *Act* which corresponds to Minsky's *agent* and Engelbart's *process*. The name is chosen because it roots each such node in its human origin rather than an abstract step.

Acts then make up *strategies* or *Recipes*. Since the most common semi-formalized activity story is the cooking recipe, I chose that name over the more abstract *strategy*.

Each *Recipe* is itself an *Act*. Thus, strategy details can be progressively elucidated if building stories top-down, or they can propagate up. That is, one can describe prepare(I,breakfast), where one of the steps is prepare(I,scrambled_eggs) and later describe the procedure of prepare(I,scrambled_eggs), then prepare(I,breakfast) will also contains the steps within prepare(I,scrambled_eggs) such as get(I,frying_pan,from_cupboard). Alternatively, one can describe prepare(I,scrambled_eggs), and when describing the general prepare(I,breakfast), the sub-steps can be taken into account.

The problem now to create a system that can collect such input from a user, and provide feedback to optimize the process.

B. User Interface

As a gateway to Augmented Intelligence, I looked into Augmented Reality[3], where much work is concentrated on making better visual input devices such as goggles that can overlay information over the perceived world. Alas, these devices are experimental and expensive. Instead, I looked back at the requirements for building a *Recipe*. The core of that is the story the user provides. And such narratives are most commonly elicited and provided via speech.

A socially acceptable, mobile, input/output audio device is the bluetooth headset. This would allow one to describe activities as well as receive instructions anywhere, anytime.

At this point in time, Speech Recognition systems such as Dragon NaturallySpeaking 10[4] and Text-to-Speech systems such as AT&T Natural Voices[5] are incredibly powerful and are able to handle understanding and producing, respectively, complex sentences with speed and accuracy.

Thus, I chose to create a prototype that works via a text-based prompt. This would allow fast development, as well as an easy transition to the voice communication model.

C. Story Comprehension

In order to enhance the quality of the data, one of the biggest pitfalls of verbal elicitation must be eliminated: humans only remember some of what they did. So, interactive narration of an activity as it is performed should provide the best accuracy.

A goal is inputted as a relatively simple sentence describing a the full *Act* to be achieved. Subsequently, each *Act* is inputted as it occurs.

Each act is analyzed, and a decision chain is constructed. Once the goal has been completed, the user is asked to rate the satisfaction this particular strategy provided. That is, how well did the strategy achieve the goal. This rating is distributed across the decision chain which is merged with any previous decision chains into a decision tree with various composite satisfaction values for each node.

D. Act Breakdown

Initially, I chose to use FrameNet[8] as the corpus of possible semantic frames[9], modeling each *Act* as a *Frame*. Frames are gestalt representations of an event, state, or object.

However, parsing sentences into frames turned out to be a fairly tricky and involved process. In order to reduce the time required to develop a prototype VerbNet was chosen instead, since it provides a representation closer to the linguistic layer despite some lingering issues in ability to infer detailed semantics[10].

IV. Implementation

The prototype was written in pure Python using NLTK's[6] VerbNet interface and Hugo Lin's MontyLingua[7] for sentence parsing.

[3] http://en.wikipedia.org/wiki/Augmented_reality
[4] http://www.nuance.com/naturallyspeaking/
[5] http://www.naturalvoices.att.com/
[6] http://nltk.org
[7] http://web.media.mit.edu/ hugo/montylingua/

A prompt requests input from a user, usually as natural language sentences. in addition the `.done` command indicates a goal is complete.

A listing of simple interaction with the program can be found in the Appendix at the end of this paper.

A. Parsing Utterances

As each utterance is entered, MontyLingua parses it and provides the verb and the arguments, e.g. in "I am making scrambled eggs", "make" is the verb while "I" and "scrambled eggs" are the arguments.

Then, the verb is located in the VerbNet corpus. A verb can map unto more than one VerbClass, so some sort of disambiguation is required. For high precision requirements with large gold standards available techniques such as maximum entropy-based tagging are available[11]. In this case, to speed up the process I chose a much simpler approach that has proved to be fairly reliable.

In the VerbNet corpus, each VerbClass contains a list of possible syntactic frames, which describe the pattern of the sentence, e.g. NP[AGENT] VERB NP[THEME]. These patterns are derived from the data and matched against the part of speech pattern in the user's utterance. The closest match is picked to be the correct VerbClass for the utterance.

Thus, a short name of the form verbclass(arg1,arg2,...) is derived, and can be used as the concise representation of an Act, e.g. run(boy,to_school) for "The boy walked to school".

B. Choosing Optimal Action

The best action at each point is chosen in a rather simplistic manner. If there is a decision tree for the current goal, it is traversed parallel to the user. After each utterance, it is compared to the possible choices available at that layer in the tree. If that choice is not rated among the top of the layer, the user is prompted to either re-do or continue. If the user ignores and proceeds, a wholly new decision chain will be generated, and the program will stop comparing the user to the existing decision tree.

V. CONCLUSION

Anecdotal testing of the system proved to be extremely promising. Despite the very simple algorithms used to speed up prototype development, the parsing was rather reliable. I was able to demonstrate all the concepts I set out to show could work.

The practical intellectual augmentation of humans is very near and much easier to achieve with current tools. Alas, the thinking of the old-style AI pioneers I have drawn upon has fallen out of fashion. Thus, I hope to show that it can still be applicable and very useful.

The day after giving a presentation about this project I had the fortune to talk to Marvin Minsky in person. We discussed this system as well as general approaches to providing machines with semantics. Minsky remarked that people hate being told what they already know, which matched a design principle I used to make my system as unobtrusive as possible, relying on user narration instead. He also insisted that one of the major failures of AI has been that machines were never imbued with goals. He pointed to an example from [7] (p. 261):

Mary was invited to Jack's party.
She wondered if he would like a kite.

A question is then posed "What is the kite for?". The answer is of course "a birthday gift for Jerry". However, to be able to answer a deep understanding of parties is required.

Minsky suggested that this can only be achieved by collecting stories from people, and then annotating them semantically. This is exactly the path I followed in my design.

Ultimately, the interconnected decision trees form a tangled web of semantic content describing human goals and solution rated for effectiveness.

A. Future Work

There is still some substantial work left to be done to make this concept viable. All build on top of the foundation demonstrated in the prototype.

Although arguments are parsed, and utterances are matched to a VerbClass, currently there is no semantic role mapping. This would be fairly simple to implement as an extension of the VerbClass matching algorithm. This would allow a rather significant new feature. Semantic role mapping for the arguments means that utterances can be mapped unto FrameNet frames as well as recastings of VerbClasses. This means that external agents can be created that understand the semantics of certain Frames. The arguments from the utterance are transmitted the agent, which can then interact with that node in the decision tree. An example may be an agent that understands the semantics of a Motion frame and is able to translate them into real-world objects, states an events. The Motion frame corresponds in part to VerbClasses such as run. Thus, the agent can take an *Act* that describes walking from point A to point B, and translate that to the actual GPS coordinates of those locations. Then the system can use that information to replace user narration when matching against the decision tree. Another possibility is to use this information in lieu or in combination with user narration in the construction of new *Recipes*.

Another enhancement would be to move from using decision trees to directed acyclic graphs. This would allow more accurate modeling of the way different *Recipes* are related to each other. It would also allow a more accurate distribution of satisfaction scores, since in the current model, if one node in the decision chain differed in the past, all subsequent nodes are treated as new and will have distinct satisfaction values. An extension of this would be tracking how important ordering relations are to satisfaction. That is, it will uncover that a strict ordering of certain of the *Acts* is essential, or perhaps that order doesn't matter to a particular goal at all, and the system should just run a checklist and make sure everything was completed.

The next step is to enable interaction with others. The most basic element is the sharing of the decision trees. This means that one can share solutions to problems with others. This means that problem solving strengths in particular areas can be spread throughout the community of users. The advanced element is that of shared goals and collaborative learning. That is, rather than training a client agent in only your own decision trees, multiple clients learn a single *Recipe* with multiple human actors in it. Then, each client is able to redeploy what has been learned in a social environment.

These last additions would allow for substantial gains in human efficiency. Participants in goal and strategy sharing can be human or machine. Thus, shallow collection of *Recipes* can lead to deep social intelligence interconnection.

REFERENCES

[1] J. R. Flynn, *What is Intelligence?: Beyond the Flynn Effect*. Cambridge University Press, August 2007.

[2] H. A. Simon, "Designing organizations for an information-rich world," in *Computers, Communication, and the Public Interest*, M. Greenberger, Ed. Martin Greenberger, 1971.

[3] D. C. Engelbart, "Augmenting human intellect: A conceptual framework," Air Force Office of Scientific Research, Stanford Research Institute, Menlo Park, CA, Summary Report AFOSR-3223 - Contract AF49(638)-1024, SRI Project 3578, October 1962.

[4] V. Bush, "As we may think," *interactions*, vol. 3, no. 2, pp. 35–46, March 1996.

[5] A. Newell, J. C. Shaw, and H. A. Simon, "Protocol analysis: Verbal reports as data," in *Proceedings of the International Conference on Information Processing*, 1959, pp. 256–264.

[6] K. A. Ericsson and H. A. Simon, *Protocol Analysis: Verbal Reports as Data*. MIT Press, 1993.

[7] M. Minsky, *Society of Mind*. Simon & Schuster, March 1988.

[8] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk, *FrameNet II: Extended Theory and Practice*, 2006. [Online]. Available: http://framenet.icsi.berkeley.edu/book/book.pdf

[9] C. J. Fillmore, "Frame semantics and the nature of language," *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, vol. 280, pp. 20–32, 1976.

[10] A. Zaenen, D. Bobrow, and C. Condoravdi, "The encoding of lexical implications in verbnet predicates of change of locations," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, E. L. R. A. (ELRA), Ed., Marrakech, Morocco, may 2008.

[11] M. Fleischman and E. Hovy, "A maximum entropy approach to framenet tagging," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 22–24.

APPENDIX

```
***** MontyLingua v.2.1 ******
**** by hugo@media.mit.edu *****
Lemmatiser OK!
Custom Lexicon Found! Now Loading!
Fast Lexicon Found! Now Loading!
Lexicon OK!
LexicalRuleParser OK!
ContextualRuleParser OK!
Commonsense OK!
Semantic Interpreter OK!
Loading Morph Dictionary!
********************************
  Lemmatiser OK!

=========
  OPAS
=========

  Please start by inputting your goal
name.
  > I am making scrambled eggs
  [o] Goal accepted: preparing(I,egg) [o]
Input the steps:
  > I am getting eggs from the fridge
  [.] Added step: get(I,egg,from_fridge)
  > I am getting bowl from the cupboard
  [.] Added step: get(I,bowl,from_cupboard)
  > I am whisking eggs in bowl
  [.] Added step: spank(I,egg,in_bowl)
  > I am getting a frying pan from cupboard
  [.] Added step: get(I,fry_pan,from_cupboard)
  > I am frying the eggs
  [.] Added step: cooking(I,egg)
  > .done
  How satisfied are you with the
result? [Unsatisfactory/So-So/Ok/Almost
Perfect/Perfect]> ok
  To continue, input another goal.
  > I am making scrambled eggs
  [o] Goal accepted: preparing(I,egg)
  [o] Input the steps:
  > I am getting eggs from the fridge
  [.] Added step: get(I,egg,from_fridge)
  > I am dancing in the streets
  [.] Added step:
  modes_of_being_with_motion(I,in_street)
  [!] Previous experience shows that
the following have been better choices:
['get(I,bowl,from_cupboard)']
  Would you like to replace the last step?
[Yes/No]>
```