FUNDAMENTAL/INFORMATIONAL
Sparx Systems Central Europe GmbH Jan van Oort, Ed.
Engineering Note: 1 (EN 1)
Category: Technical Documentation
Updates and completes: (krino white paper)

The Foundation of the *krino* Artificial Intelligence

Abstract

As *krino* was originally an invention by a single person, joined later by scientists, engineers, and business men, development and business continuity MUST now be a core concern of the *krino* team. To this end, this document describes, at a technical level of computer science, programming, linguistics, philosophy, and mathematics, the various steps and operations needed for *krino* to become a working, inspectable, verifiable, and general-purpose AI.

This is the first edition of this document. It updates (krino white paper).

Status of this Engineering Note

This document is a product of *krino's* original inventor. It is currently in preparation, and has not yet been inspected or reviewed by other *krino* team members.

Copyright Notice

1. Introduction and Background

1.1 Introduction

This document describes the individual components, or building blocks, necessary for building a natural language-based, inspectable, and verifiable Artificial Intelligence. As such, it is concerned with scientific frameworks and technologies, at the level of “what”, decidedly not at the level of “how”. That is: we describe a technology X, or a capability Y, needed to produce such an AI as a working computing artifact; we do not specify that X should be implemented in programming language A, or that capability Y should be realized based upon framework B or C. Also, this document makes no assumptions about the feasibility of constructing such components and building blocks, although we do recognize that for at least some of these we are at the very edge of what is currently (technically) possible and (scientifically) known.

1.2 Background

The goal of the krino project is the production of an Artificial Intelligence that exhibits defining features currently not found, to the knowledge of the author, in any extant AI:

a. purely natural language-based reasoning and analysis capabilities

b. two-way interaction with humans in natural language

c. full inspectability and verifiability, meaning that when the AI makes a recommendation, draws a conclusion, or states any proposition, its internals can be inspected as to how and why such propositions, conclusions, or recommendations came into being

d. capability to extend the knowledge perimeter on a topic by ingesting additional sources

e. capability to detect causality, causal nexuses, and causal statements in ingested text

f. capability to detect arguments and argument types used in the ingested text

Decidedly, as stated in [HAU 15], "*the aim is not to build superlogical or inhumanly rational automata. Our human tendencies to take things for granted, make intuitive leaps, and resist implausible conclusions are not weaknesses that AI strives to overcome but abilities integral to real intelligence that AI strives to share*". The reason we push for inspectable and verifiable AI is well expressed by Rule 1 and Rule 5, part of "The Rules" established under "Moral Responsibility for Computing Artifacts" by the Ad Hoc Comittee for Responsible Computing, and rendered thusly in [BRINKMAN 13]:

> (Rule 1) *People who make computing artifacts are morally responsible for the foreseeable effects of those artifacts*.
>
> (Rule 5) *People who desing computing artifacts must provide enough information about them so that others can foresee the consequences of using them*.

Moreover, the reason for which we choose natural language as the basis for the AI's reasoning and analysis capabilities, is twofold:

> a. all human knowledge can be represented in natural language
>
> b. handling such knowledge is subject to the two semantic processes known as denotation and connotation, as expressed in [ECO 76], p. 55 *sqq*. Our effort is highly novel in that there is no currently extant AI capable of reasoning at the connotational level.

Reason b. merits to be expounded somewhat. In the first decade of the 21st century, two strains of AI were commonly identified: *symbolic AI* and *subsymbolic AI*, also known as being included in the field of *soft computing*. According to [RALSTON 00], the basic premises of soft computing are:

> a. imprecision and uncertainty are pervasive

b. precision and certainty carry a cost

with the guiding principle of soft computing being:

"exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost". Although initially neural networks and its encompassing field, known as *neurocomputing*, were classified as part of soft computing, neural networks — now the world's overwhelmingly predominant type of AI — have taken the route of fully, or nearly fully, statistics-based reasoning, which resulted in AIs that are neither inspectable nor verifiable.

Resuming, we see that the various motivations, ethical considerations, desired features, and necessary AI capabilities "interlock", i.e. each of them is influenced or rationalized by at least one of the others. This makes the actual task of building an AI like krino a rather challenging one.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. This document uses, where necessary or useful, those keywords specified in [RFC2119]; where they appear in UPPERCASE, they are implicitly meant to reference these words, with their definition and force, in said RFC. Where they appear in lowercase, there is no implicit reference to RFC 2119.

3. Storage

4. Linguistics

The krino AI works at the *phrase* (or sentence) level as its basic unit of ingestion. This means that any text ingested is cut down into phrases, parsed as phrases and stored as phrases. This does not mean that individual words are not looked up, or stored: they are. However, the basic unit of work provided to the causal and argumentation levels is the phrase. Therefore:

R 4.1 krino MUST produce a parse tree of a well-formed natural language phrase.

R 4.2 The scientific theory and framework used to parse natural language phrases MUST be the one specified in [GOBBO 11].

R 4.3 krino MUST be able to reconstitute a well-formed natural language phrase after having parsed and stored it.

R 4.4 krino MAY choose to store the parse trees, corresponding to well-formed natural language phrases and produced as per R 4.1, in its storage medium.

Arguments consist of phrases, and a discourse consists — among other things — of arguments. Therefore:

R 4.5 arguments detected by krino MUST be shown to consist of individually identifiable phrases, which as per R 4.3 can be reconstituted in their original form.

R 4.6 in a way entirely parallel to R 4.5, a discourse ingested by krino MUST be shown to consist of individually identifiable arguments.

Natural language is, by its nature, ambiguous; hence, freely citing [RALSTON 00] (p. 1218), every software artifact parsing and handling natural language will encounter the following forms of ambiguity:

a. simple lexical ambiguity (e.g. “duck” can be be a noun [the animal] or a verb [to avoid something thrown or shot])

b. structural or syntactic ambiguity (e.g. in “I saw the man with a telescope”, the telescope might be used for the viewing or might be held by the man being observed).

c. semantic ambiguity (e.g. “go” as a verb has over 10 distinct meanings in any dictionary).

d. pragmatic ambiguity (e.g. “Can you lift that rock?” may be a yes/no question or a request to lift the rock).

e. referential ambiguity (e.g. in “Jack met Sam at the station. He was feeling ill....”, it is not clear who is ill, although the remainder of the sentence viz. the discourse’s context might suggest a preferred interpretation).

f. logical fallacies arising from one of the above types of ambiguity interfering with standard logical inference (e.g., in “Nothing is bettern than eternal bliss. Tepid tea is better than nothing. Hence, tepid tea is better than eternal bliss”, the culprit is the very ambiguous word “nothing”, used together with the transitivity of “better than”.)

Therefore:

R 4.7 krino MUST demonstrably resolve, before any further parsing of a natural language phrase is carried out, the lexical ambiguities present in the phrase.

R 4.8

5. Causality

6. Arguments and Argumentation

7. Security Considerations

8. References

8.1 Normative References

8.2 Informative References

[BRINKMAN 13] Bo Brinkman and Alton F. Sanders, “Ethics in a Computing Culture”, Cengage Learning, 2013, p. 327

[ECO 76] Umberto Eco, “A Theory of Semiotics”, Indiana University Press, 1976

[GOBBO 11] Federico Gobbo and Marco Benini, “Constructive Adpositional Grammars. Foundations of Linguistics”, Cambridge Scholars Publishing, 2011

[HAU 15] John Haugeland, “*Artificial Intelligence*” (article), in: Robert Audi (ed.), “The Cambridge Dictionary of Philosophy”, Cambridge University Press, 2015, p. 63

[RALSTON 00] Anthony Ralston, Edwin D. Reilly and David Hemmendinger (eds.), “Encyclopedia of Computer Science”, Nature Publishing Group, 2000.

Acknowledgements

Contributors

Author’s Address