Lojban++: An Interlingua for Communication between Humans and AGIs

Ben Goertzel

Novamente LLC

Abstract. Lojban is a constructed language based on predicate logic, with no syntactic ambiguity and carefully controllable semantic ambiguity. It originated in the middle of the last century, and there is now a community of several hundred human speakers. It is argued here that Lojban++, a minor variation on Lojban, would be highly suitable as a language for communication between humans and early-stage AGI systems. Software tools useful for the deployment of Lojban++ in this manner are described, and their development proposed.

1 Introduction

Human “natural language” is, generally speaking, unnatural to AGI systems. The only exceptions would be AGI systems constituting very close emulation of human brains or minds; and the pursuit of such systems is not currently central to the AGI field. Yet, natural or not, understanding of human language is obviously critical to any AGI system that wants to interact flexibly in the human world, and/or that wants to ingest the vast corpus of knowledge that humans have created and recorded.

The problem of endowing AGI systems with understanding of human language, and human commonsense knowledge, presents a bit of a ”chicken and egg” problem. If an AGI understood human language effectively, it could pick up a lot of commonsense knowledge via reading. But if an AGI had strong commonsense knowledge, it could much more straightforwardly be made to understand human language effectively (since computational linguistics technology already enables fairly powerful syntax parsing, the main obstacle in the way of AGIs understanding human language is commonsense understanding such as is required for practical semantic interpretation).

How can this deadlock be broken? The obvious paths involve developing either computational linguistics or commonsense reasoning far enough, so that advances in one area will drive advances in the other. But here we will propose a more innovative and eccentric solution: the use of the constructed language Lojban (or, more specifically, its variant Lojban++), which occupies an interesting middle ground between formal languages like logic and programming languages, and human natural languages. A complete grammar of Lojban is presented and explained in [1], along with significant discussion of Lojban semantics and pragmatics; further information, including tutorials, is presented online and linked

K.-U. Kühnberger, S. Rudolph, and P. Wang (Eds.): AGI 2013, LNAI 7999, pp. 21–30, 2013.

from `http://lojban.org`. While there is no published hard-copy Lojban dictionary, there is a website `jbovlaste.lojban.org/` that serves this purpose and that is frequently updated as new coinages are created and approved by the Logical Language Group, a standing body charged with the maintenance of the language.

We argue here that communicating with AGI systems in Lojban++ may provide a way of

- providing AGI systems with experientially-relevant commonsense knowledge, much more easily than via explicitly encoding this knowledge in logic
- teaching AGI systems natural language much more quickly than would otherwise be possible, via communicating with AGIs in parallel using natural language and Lojban++

To put it more precisely: the essential goal of Lojban++ is to constitute a language for efficient, minimally ambiguous, and user-friendly communications between humans and suitably-constructed AI software agents such as early-stage AGI systems. Another way to think about the Lojban++ approach is that it allows an AGI learning/teaching process that dissociates, to a certain extent, "learning to communicate with humans" from "learning to deal with the peculiarities of human languages." Similar to Lojban on which it is based, Lojban++ may also be used for communication between humans, but this interesting possibility will not be our focus here.

Some details on the particulars of the Lojban++ language proposal, aimed at readers familiar with Lojban, are given in [2], available online at [1]. In this paper we describe Lojban++ and related ideas at a more abstract level, in a manner comprehensible to readers without prior Lojban background.

We will focus here on the textual version of Lojban; however, there is also a spoken version, carefully designed to minimize acoustic ambiguity and make speech-to-text easier than with natural languages.

2 Lojban versus Lojban++

Lojban is itself an outgrowth of another constructed language, Loglan, created by Dr. James Cooke Brown around 1955 and first widely announced in a 1960 Scientific American article [3]. Loglan is still under development but now is not used nearly as widely as Lojban. First separated from Loglan in 1987, Lojban is a constructed language that lives at the border between natural language and computing language. It is a "natural-like language" in that it is speakable and writeable by humans and may be used by humans to discuss the same range of topics as natural languages. Lojban has a precise, specified formal syntax that can be parsed in the same manner as a programming language, and it has a semantics, based on predicate logic, in which ambiguity is carefully controlled. Lojban semantics is not completely unambiguous, but it is far less ambiguous

[1] `http://www.goertzel.org/papers/lojbanplusplus.pdf`

than that of any natural language, and the careful speaker can reduce ambiguity of communication almost to zero with far less effort than in any natural language. On the other hand, Lojban also permits the speaker to utilize greater ambiguity when this is desirable in order to allow compactness of communication.

Many individuals attempting to learn and use Lojban have found, however, that it has two limitations. The Lojban vocabulary is unfamiliar and difficult to learn - though no more so than that of any other language belonging to a language family unfamiliar to the language learner. And, more seriously, the body of existing Lojban vocabulary is limited compared to that of natural languages, making Lojban communication sometimes slow and difficult. When using Lojban, one must sometimes pause to concoct new words (according to the Lojban principles of word construction), which can be fun, but is much like needing to stop over and over to build new tools in the context of using one's toolkit to build something; and is clearly not optimal from the perspective of teaching AGI systems.

To address these issues, Lojban++ constitutes a combination of Lojban syntax and Lojban vocabulary, extended with English vocabulary. So in a very rough sense, it may perhaps be understood as a pidgin of Lojban and English. Lojban++ is less elegant than Lojban but no more difficult (and perhaps easier) to learn, and much easier to use in domains to which Lojban vocabulary has not yet been extended. In short, the goal of Lojban++ is to combine the mathematical precision and pragmatic ontology that characterize Lojban, with the usability of a natural language like English with its extensive vocabulary.

Although Lojban has not been adopted nearly as widely as Esperanto (an invented language with several hundred thousand speakers), the fact that there is a community of several hundred speakers, including several dozen who are highly fluent at least in written Lojban, is important. The decades of communicative practice that have occurred within the Lojban community have been invaluable for refining the language. This kind of practice buys a level of maturity that cannot be obtained in a shorter period of time via formal analysis or creative invention. For example, the current Lojban treatment of quantifiers is arguably vastly superior to that of any natural language [1], but that was not true in 1987 when it excelled more in mathematical precision than in practical usability. The current approach evolved through a series of principled revisions suggested from experience with practical conversation in Lojban. Any new natural-like language that was created for human-AGI or AGI-AGI communication would need to go through a similar process of iterative refinement through practical use to achieve a similar level of usability.

3 Some Simple Examples

Now we give some examples of Lojban++. While these may be somewhat opaque to the reader without Lojban experience, we present them anyway just to give a flavor of what Lojban++ looks like; it would seem wrong to leave the discussion purely abstract. The reader who is curious to understand Lojban++ better is encouraged to look at the language specification at the URL given above.

Consider the English sentence,

When are you going to the mountain?

When written in Lojban, it looks like:

do cu'e klama le cmana

In Lojban++, with the judicious importation of English vocabulary, it takes a form more recognizable to an English speaker:

you cu'e go le mountain

A fairly standard predicate logic rendition of this, derived by simple, deterministic rules from the Lojban++ version, would be

```
    atTime(go(you, mountain), ?X)
```

Next, consider the more complex English sentence,

When are you going to the small obsidian mountain?

In Lojban, there is no word for obsidian, so one needs to be invented (perhaps by compounding the Lojban words for "glass" and "rock," for example), or else a specific linguistic mechanism for quoting non-Lojban words needs to be invoked. But in Lojban++ one could simply say,

you cu'e go le small obsidian mountain

The construct "small obsidian mountain" is what is called a Lojban tanru, meaning a compound of words without a precisely defined semantics (though there are recognized constraints on tanru semantics based on the semantics of the components [4]). Alternatively, using the Lojban word, marji, which incorporates explicit place structure (x1= material/stuff/matter of composition x2), a much less ambiguous translation is achieved:

you cu'e go le small mountain poi marji loi obsidian

in which "poi marji loi obsidian" means "that is composed of [a mass of] obsidian." This illustrates the flexible ambiguity achievable in Lojban. One can use the language in a way that minimizes ambiguity, or one can selectively introduce ambiguity in the manner of natural languages, when desirable.

The differences between Lojban and Lojban++ are subtler than it might appear at first. It is key to understand that Lojban++ is not simply a version of Lojban with English character-sequences substituted for Lojban character-sequences. A critical difference lies in the rigid, pre-determined argument structures associated with Lojban words. For instance, the Lojban phrase

klama fi la .atlantas. fe la bastn. fu le karce corresponds to the English phrase

that which goes from Atlanta to Boston by car

To say this in Lojban++ without using “klama” would require

go fi’o source Atlanta fi’o destination Boston fi’o vehicle car

which is much more awkward. On the other hand, one could also avoid the awkward Lojban treatment of English proper nouns and say

klama fi la Atlanta fe la Boston fu le car

or

klama fi la Atlanta fe la Boston fu le karce

It’s somewhat a matter of taste, but according to ours, the latter most optimally balances simplicity with familiarity. The point is that the Lojban word “klama” comes with the convention that its second argument (indexed by “fi”) refers to the source of the going, its third argument (indexed by “fe”) refers to the destination of the going, and its fifth argument (indexed by “fu”) refers to the method of conveyance. No such standard argument-structure template exists in English for “go”, and hence using “go” in place of “klama” requires the use of the “fi’o” construct to indicate the slot into which each of the arguments of “go” is supposed to fall.

The following table gives additional examples, both in Lojban and Lojban++.

English	I eat the salad with croutons
Lojban	mi citka le salta poi mixre lo sudnabybli
Lojban++	mi eat le salad poi mixre lo crouton mi eat le salad poi contain lo crouton
English	I eat the salad with a fork
Lojban	mi citka le salta sepi’o lo forca
Lojban++	mi eat le salad sepi’o lo fork
English	I will drive along the road with the big trees
Lojban	mi litru le dargu poi lamji lo barda tricu
Lojban++	mi ba travel fi’o vehicle lo car fi’o route le road poi adjacent lo so’i big tree mi ba litru fi lo car fe le road poi adjacent lo so’i big tree mi ba drive fi’o route le road poi adjacent lo so’i big tree
English	I will drive along the road with great care
Lojban	mi litru le dargu ta’i lo nu mi mutce kurji
Lojban++	mi ba drive fi’o route le road ta’i lo nu mi much careful mi ba litru le road ta’i lo nu mi much careful

English	I will drive along the road with my infrared sensors on
Lojban	mi ba litru le dargu lo karce gi'e pilno le miktrebo'a terzga
Lojban++	mi litru le road lo car gi'e use le infrared sensor mi litru le road lo car gi'e pilno le infrared te zgana mi drive fi'o vehicle lo car fi'o route le road gi'e use le infrared sensor

English	I will drive along the road with the other cars
Lojban	mi litru le dargu fi'o kansa lo drata karce
Lojban++	mi ba drive fi'o route le road fi'o kansa lo so'i drata car mi ba drive fi'o route le road fi'o with lo so'i drata car mi ba litru le road fi'o kansa lo so'i drata car

4 The Need for Lojban Software

In order that Lojban++ be useful for human-AGI communications, parsing and semantic mapping software need to be produced for the language, building on existing Lojban software.

There are a number of fully functional Lojban parsers, see `http://www.lojban.org/tiki/Dictionaries%2C+Glossers+and+parsers`. The creation of a Lojban++ parser based on the existing Lojban parser, is a necessary and a relatively straightforward though not trivial task.

On the other hand, no software has yet been written for formal semantic interpretation ("semantic mapping") of Lojban expressions - which is mainly because Lojban has primarily been developed as an experimental language for communication between humans rather than as a language for human-AGI communication. Such semantic mapping software is necessary to complete the loop between humans and AI reasoning programs, enabling powerful cognitive and pragmatic interplay between humans and AGI's. For Lojban++ to be useful for human-AGI interaction, this software must be created and must go in both directions: from Lojban++ to predicate logic and back again. As Lojban++ is a superset of Lojban, creating such software for Lojban++ will automatically include the creation of such software for Lojban proper.

There promises to be some subtlety in this process, but not on the level that's required to semantically map human language. What is required to connect a Lojban++ parser with an English-language NLP framework , for example, would be a mapping between

- the Lojban cmavo (structure word) and the argument-structure of lojban gismu (root word)
- an ontology of standardized English verbs (with subcategorization frames) and prepositions. For instance, this could be done using FrameNet [5], augmented with a handful of other prepositional and logical relationships (e.g. for dealing with space, time and inheritance)

These mappings would have to be built by hand, which would be time-consuming, but on the order of man-weeks rather than man-years of effort.[2] Once this is done, then Lojban++ can be entered into AGI essentially *as English would be if one had a perfectly functional natural language parsing and semantic understanding system.* The difficulties of human language processing will be bypassed, though still – of course – leaving the difficulties of commonsense reasoning and contextual interpretation.

For example, the Lojban root word *klama* is defined as

x1 comes/goes to destination x2 from origin x3 via route x4 using means/vehicle x5.

This corresponds closely to the FrameNet frame Motion, which has elements

- Theme (corresponding to x1 in the above Lojban definition)
- Source (x2)
- Goal (x3)
- Path (x4)
- Carrier (x5)

The Motion FrameNet frame also has some elements that klama lacks, e.g. Distance and Direction, which could of course be specified in Lojban using explicit labeling.

5 Lojban and Inference

Both Lojban and Lojban++ can be straightforwardly translated into predicate logic format (though the translation is less trivial in the case of Lojban++, as a little bit of English-word disambiguation must be done). This means that as soon as Lojban++ semantic mapping software is constructed, it will almost immediately be possible for AGI systems to reason about knowledge communicated to them in Lojban. This aspect of Lojban has already been explored in a preliminary way by Speer and Havasi's [6] JIMPE software application, which involves a semantic network guiding logical reasoning, Lojban parsing and Lojban language production. While JIMPE is a relatively simplistic prototype application, it is clear that more complex example of Lojban-based artificial inference are also relatively straightforwardly achievable via a conceptually similar methodology.

An important point to consider in this regard is that Lojban/Lojban++ contains two distinct aspects:

1. an ontology of predicates useful for representing commonsense knowledge (represented by the Lojban cmavo along with the most common Lojban content words)

[2] Carrying out the following mapping took a few minutes, so carrying out similar mappings for hundreds of different verbs should take no more than a couple weeks of effort for a suitably trained individual.

2. a strategy for linearizing nested predicates constructed using these cmavo into human-pronounceable and -readable strings of letters or phonemes.

The second aspect is of no particular value for inference, but the first aspect is. We suggest that the Lojban++ ontology provides a useful framework for knowledge representation that may be incorporated at a fundamental level into any AI system that centrally utilizes predicate logic or a similar representation. While overlapping substantially with FrameNet, it has a level of commonsensical completeness that FrameNet does not, because it has been refined via practice to be useful for real-world communication. Similarly, although it is smaller than Cyc, it is more judiciously crafted. Cyc contains a lot of knowledge not useful for everyday communication, yet has various lacunae regarding the description of everyday objects and events – because no community has ever seriously tried to use it for everyday communication.

5.1 Lojban versus Predicate Logic

In the context of Lojban++ and inference, it is interesting to compare Lojban++ formulations with corresponding predicate logic formulations. For example, consider the English sentence

Hey, I just saw a bunch of troops going into the woods. What do you want me to do?

translates into the Lojban

ju'i do'u mi pu zi viska lo nu so'i lo sonci cu nenkla le ricfoi .i do djica lo nu mi mo

or the Lojban++

Hey do'u mi pu zi see lo nu so'i lo soldier cu enter le forest .i do want lo nu mi mo

which literally transcribed into English would be something like

Hey! [vocative terminator] I [past] [short time] see an event of (many soldiers enter forest). You want event (me what?)

Omitting the "hey," a simple and accurate predicate logic rendition of this sentence would be

$$\begin{gathered} \mathit{past}(\$X) \wedge \mathit{short\_time}(\$X) \wedge (\$X = \mathit{see}(\mathit{me}, \$Y)) \wedge \\ (\$Y = \mathit{event}(\mathit{enter}(\$Z, \mathit{forest}))) \wedge \mathit{soldier}(\$Z) \wedge \mathit{many}(\$Z) \wedge \\ \mathit{want}(\mathit{you}, \mathit{event}(?W(\mathit{me})) \end{gathered}$$

where ?W refers to a variable being posed as a question be answered, and X and so forth refer to internal variables. The Lojban and Lojban++ versions have the same semantics as the predicate logic version, but are much simpler to speak, hear and understand due to the lack of explicit variables.

6 Discussion

Hopefully the above exposition of Lojban++, though incomplete, was sufficient to convince you that teaching "infant-level" or "child-level" AGIs about the world using Lojban++ would be significantly easier than teaching doing so using English or other natural languages. The question then is whether this difference makes any difference.

One could counter-argue that, if an AGI were smart enough to really learn to interpret Lojban++, then it would be smart enough to learn to interpret English as well, with only minor additional effort. In sympathy with this counter-argument is the fact that successfully mapping Lojban++ utterances into predicate logic expressions, and representing these predicate logic expressions in an AI's knowledge base, does not in itself constitute any serious "understanding" of the Lojban++ utterances on the part of the AI system.

However, this counter-argument ignores the "chicken and egg problem" of commonsense knowledge and language understanding. If an AGI understands natural language then it can be taught human commonsense knowledge via direct linguistic instruction. On the other hand, it is also clear that a decent amount of commonsense knowledge is a prerequisite for adequate natural language understanding (for such tasks as parse selection, semantic disambiguation and reference resolution, for example). One response to this is to appeal to feedback, and argue that commonsense knowledge and linguistic understanding are built to arise and grow together. We believe this is largely true, and yet that there may also be additional dynamics at play in the developing human mind that accelerate the process, such as inbuilt inductive biases regarding syntax. In an AGI context, one way to accelerate the process may be to use Lojban++ to teach the young AGI system commonsense knowledge, which then may help it to more easily penetrate the complexities and ambiguities of natural language.

This assumes, of course, that the knowledge gained by the system from being instructed in Lojban++ is genuine knowledge rather than merely empty, ungrounded tokens. For this reason, we suggest, one viable learning project may be to teach an AGI system using Lojban++ in the context of shared embodied experience in a real or simulated environment. This way Lojban++ expressions may be experientially grounded and richly understood, potentially allowing commonsense knowledge to form in an AGI's knowledge base, in a way that can be generalized and utilized to aid with various learning tasks including learning natural language.

Another interesting teaching strategy may be to present an AGI system with semantically roughly equivalent English and Lojban sentences, especially ones that are pertinent to the system's experiences. Since the system can interpret the Lojban sentences with minimal ambiguity (especially by using the experienced context to reduce any ambiguities remaining after parsing, due to tanru), it will then know the correct interpretation of the English sentences, which will provide it with very helpful "training data" that it can then generalize from to help it understand other English sentences.

Lojban and Lojban++, like any languages suitable for human communication – but even more so due to their dual roles as human and formal languages – are flexible tools that can be used in many ways. The precise best way to use these tools to advance AGI will only emerge via practice. However, we feel confident that, with suitable Lojban/Lojban++ software in hand, these languages could form a valuable addition to AGI field's armamentarium.

References

1. Cowan, J.W.: The Complete Lojban Language. The Logical Language Group (1997)
2. Goertzel, B.: Lojban++: An efficient, minimally ambiguous, user-friendly natural-like language for human-computer, computer-computer and human-human communication (2006), http://www.goertzel.org/papers/lojbanplusplus.pdf
3. Brown, J.C.: Loglan. Scientific American (June 1960)
4. Nicholas, N.: Lojban as a machine translation interlanguage. In: Fourth Pacific Rim International Conference on Artificial Intelligence: Workshop on 'Future Issues for Multilingual Text Processing' (2006)
5. Baker, C., Fillmore, C., Lowe, J.: The berkeley framenet project. In: Proc. of the COLING-ACL (1998)
6. Speer, R., Havasi, C.: Meeting the computer halfway: Language processing in the artificial language lojban. In: Proceedings of MIT Student Oxygen Conference (2004), http://sow.lcs.mit.edu/2004/proceedings.html