**Truenumbers, the natural data language for humans and AI**

Data isn’t only a technical artifact - it’s a linguistic one. Humans think in language, speak in language, and reason through language. Every dataset is a shadow of human expression, and the closer our representations align with natural language, the more usable, interpretable, and interoperable they become. This isn’t just about readability. It’s about cognition. By us—and increasingly by large language models (LLM). When data is structured in ways that mirror how we speak and reason, it becomes more accessible to both humans and machines.

*Truenumbers* are data values expressed as atomic sentences, each with a subject phrase (the entity of interest); a property phrase (an attribute of the subject) and the quantitative or qualitative data value itself. Some examples:

The Earth has approximate diameter = 8000 miles

Hyundai Ioniq 5 EV has customer rating = excellent

Tokyo has population = 13.5 million

These linguistic units of meaning are parseable, composable, and context aware. They bridge the gap between structured data and natural language, making them ideal for LLM ingestion, retrieval-augmented generation (RAG), and semantic search.

Schematic data has evolved primarily as a platform for computation, with meaning only weakly hinted at by column labels or class names. In contrast, natural language is the essence of meaning for us, but intractible as a platform for computation.

Along with this change, came the belief that data which is selected, “cleaned” and aligned to a schema or ontology has been improved from its original state. This is true from the perspective of the developers and software intending to work with that data, but the opposite is true from the perspective of information capture. There is no richer, higher quali

Truenumber data inherits these benefits.

We start with the observation that SQL fields, XML elements, classes, attributes and so forth are connected to the real world *only* by the labels we put on them. Consider the SQL query

**SELECT FerryRng from Aircraft WHERE Type = ‘FA-18E’**

It’s operational meaning is specified by SQL and is exactly the same as for the query

**SELECT Col1 from Table1 WHERE Col2 = ‘value’**

We *understand* the first query in domain terms because “Aircraft”, “Type” and “F-18E” mean something to us, and we can guess that “FerryRng” means “Ferry Range” which we also understand.

In human terms, what we are asking for is “the ferry range of an F-18E aircraft”. In a reference book we don’t compose a query, we look for an entry in the index at the back

Aircraft, FA-18E

- ferry range . . . . p. 233

so, one wonders, why not store and index data in a similar way, by the phrases people use to describe things?

FA-18E aircraft **has** ferry range **=** 1660 nmi

This simple sentence uses a keyword “has” to separate the property of interest “ferry range” from “FA-18E aircraft”, the subject having that property. The “=” indicates that “1800 nmi” is the value of that property.

where the subject FA-18E is distinct from the property ferry range, and units of measure are given as part of the value. This kind of data can be discovered by searching on property terms like ferry range, or FA-18E used in a subject phrase.

The language principle:

Understanding and proper use of data can only result from narrative documentation

Communities of interest - warfighters and system architects - must share a common understanding and general terminology for things to work. Labeling a column FerRngNm does not change this, or somehow make information more precise. It only puts phrases like ferry range in the background, inaccessible to systems and users. The more we use structured phrases to describe information, the more precise our descriptions will become, and the more our systems and analytics can benefit from them.

**Data as a common language**

Truenumbers sees data as that part of language we use to describe information in a systematic way. For example, when you say “this aircraft carries a crew of 5”. But description of information is not a priority for schema-based data. The label **seaTmpF** with description “sea temperature measurements in Fahrenheit” is called *meta*data because it’s not part of the data itself. Truenumbers takes a different approach, making description the main content of data, like a reference book.

Books have a table of contents and index in the back for accessing content, and we learn a lot from browsing them.

To find sea temperature, we look in the back of the book:

**Sea, Caspian, temperature of, p236–7; salinity of, p178, p232**

It makes sense to store data in a similar way:

**Sea, Caspian, temperature of, 52F; salinity of, 35 g/kg**

Truenumbers uses a structured subset of natural language, the main component of which is the *noun phrase*, like "Caspian sea*"*. In addition to adjective modifiers, we allow the keyword *of* as a possessive association, for example "radar antenna of F-22 aircraft 243*"*.

This simple structure for noun phrases lets them be represented as path-like strings, with colon and slash to denote word sequence, and “of”. So the phrase *radar antenna of F-22 aircraft 243* can be written as *243:aircraft:F-22/antenna:radar*. These *phrase paths* are ideal for use as indexes, and for search.

With this simple grammar for noun phrases, we can build a truenumber statement in either of two equivalent forms:

**<subject phrase> has <property phrase> = <value>**

or

**<property phrase> of <subject phrase> is <value>**

Values can be numbers with units of measure and tolerance, quoted strings, or noun phrases. Here are some examples:

**Caspian sea has salinity = 35 +/- 5 g/kg**

**security status of Logan airport is red alert**

**NAVWAR has mission statement = “*To identify, develop, deliver and sustain information warfare capabilities and services that enable naval, joint, coalition and other national missions.”***

**Language and computation**

Why is this kind of data representation important? We're familiar with data in tables, XML, JSON, RDF, all sorts of formats - so why do we need another? The difference is subtle, but profound, similar to the difference between say, a program written in Java and a white-paper written in WORD. It's true that they are just documents in two different languages, but the Java program is written to be interpreted by a computer. The paper is written for people. This distinction is obviously huge. All of human knowledge and collaboration is based in narrative language. You can't write a book like the Art of War in Java, and you can't compute the trajectory of a missile in WORD. Computers have always reflected this distinction. They help write, edit, manage and search language they can't understand; or they can do computations people can use.

Computation generally involves *data* *-* discrete bits of information derived from the real world, and configured in the computer for processing. It isn't narrative for people to read. Truenumbers is the first data representation to be based on human language with a narrative purpose. It is able to be written, managed and searched like any narrative text, but being individual bits of information, not a lengthy book or white paper, truenumbers can support computation as well or better than any of the formats we're used to.

**What problems does this solve?**

In the world of complex systems we face difficulties associated with computational data. Silos with different schemas, and even different types of schemas limit agility, flexibility and interoperability. More precise data models (e.g. ontologies) are extremely hard to build and maintain, and move farther from the expressivity and nuance of language. Integration and federation across multiple schemas and domains are ongoing tasks that never seems to catch up to the systems being developed and integrated.

But natural language doesn't have these problems. We can speak and write about anything without limitation, and when assembling lots of narrative, we are able to negotiate common ground in terminology and structure pretty easily. Truenumber data inherits these benefits.

--

Tables are the most common kind of schematic data, but as explained earlier, they don’t say much. To represent their contents in truenumbers requires incorporation of information from data dictionaries or external sources

These tables and keys encompass identification of squadrons, aircraft and equipment, their configuration, status and relationships. They represent the most recently known state of these things.

**operational status of FA-18F modex 111 is DOWN**

**@ 2021-06-02 OOMA update**

**operational status description of FA-18F modex 111 is “TANK 2 / ALQ-214”**

**@ 2021-06-02 OOMA update**

The above is two truenumbers, each with a tag.