Natural Language and Intelligence: A Fundamental Approach

Abstract

Despite thousands of years of philosophy and 60 years of research on Artificial Intelligence, the scientific definition of intelligence is still abstract, and therefore difficult to implement in software. This should raise the question whether AI can be researched without a clear understanding of the fundamentals of intelligence itself.

If the foundation of a scientific discipline is right, it integrates all its sub-disciplines. However, in the field of Artificial Intelligence the sub-disciplines are incompatible. This proves the scientific approach to AI wrong.

This paper describes a fundamental approach to implement Artificial Intelligence through natural language.

Introduction

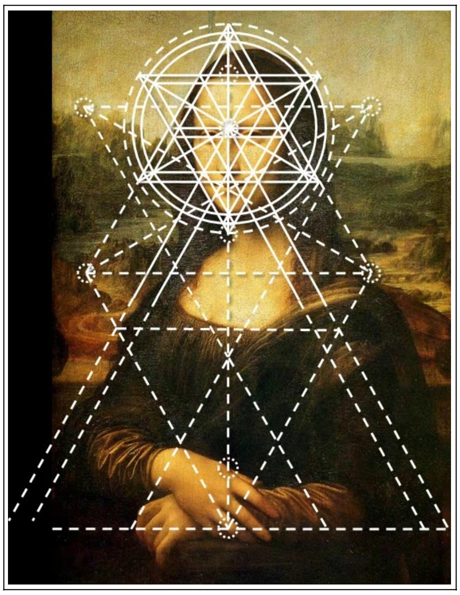
According the general consensus, natural language is too fluid and too complex to be processed by software into depth. Therefore, natural language is processed top-down – “as good as it gets” – lacking a deeper level of understanding, unable to touch the deeper meaning.

In order to fix the lack of understanding, semantics is added to information systems artificially, in the hope those systems will grasp the deeper meaning. Unfortunately, adding artificial semantics might not solve the problem, if the root of the problem lies deeper than that: in understanding intelligence itself.

A correct procedure to include Artificial Intelligence to information systems:

- **Analysis**: Intelligence must be defined in an unambiguous, fundamental (=natural[1]) and deterministic (=implementable) way, because without such a fundamental relationship, any attempt to add AI to information systems will surely fail;

- **Design**: Methods must be developed to implement that unambiguous, fundamental and deterministic definition of intelligence in software.



Hints for fundamental research:

- Grammar gives structure to natural language, in a similar way as Leonardo da Vinci gave structure to his Mona Lisa;
- Children have no problem in deriving the logic behind grammar, because it is hard-wired in our brains;
- Find that logic contained within grammar, and it will provide the foundation for AI through natural language.

1 Natural rather than artificial. The system itself might be artificial, but not its foundation.

1 Back to basics: What is intelligence?

Without being exhaustive, an unambiguous, fundamental (=natural) and deterministic (=implementable) definition of intelligence is used:

Intelligence is the capability of autonomously *associating*, *discriminating*, *learning*, *planning* and *predicting*, with the aim to reach a predefined goal.

More into detail:

- *associating or (combining)* of individual or separate objects, with the aim of achieving a goal that can not be achieved by either of those objects separately;
- *discriminating (or differentiating)* compound or intertwined objects, with the aim to clarify the situation, by putting them in their own context;
- *learning* is using knowledge and experience to differentiate successes from mistakes;
- *planning* future actions to achieve a goal;
- *predicting* possible consequences of those planned actions.

Initially, the capabilities *associating* and *discriminating* will be implemented, see 3.3 Semantics.

2 A fundamental relation with intelligence

Science is about researching fundamentals, like appearances of magnetism were researched in order to understand magnetism and appearances of electricity were researched in order to understand electricity. In the same way, the appearances of intelligence should be researched in order to understand intelligence.

Natural language is an expression of intelligence, because intelligence is required to form sentences in natural language. To able to research appearances of intelligence expressed in natural language, an unambiguous, fundamental (=natural) and deterministic (=implementable) structure is required: grammar.

Without being exhaustive, the following fundamental path from grammar to intelligence is used to implement Artificial Intelligence through natural language:

- Grammar provides **structure** to a language;
- Structure is a result of **logic**;
- Logic is a result of intelligence.

Find that logic contained within grammar,
and it will provide the **foundation** for Artificial Intelligence through natural language.

3 Design

Some techniques to implement fundamental AI are described below.

3.1 Grammar rules

Traditionally, all sentences are accepted during the NLP analysis. Disadvantage: It is not clear how deep the knowledge is processed.

By implementing grammar rules, by only accepting sentences according this (limited) grammar and by analyzing them in a fundamental way, the depth of the system is clear: All accepted sentences are fully “understood”. How more depth can be accomplished, will be explained below.

3.2 The knowledge structure

The analyzed sentences are stored in a knowledge structure, which is designed to guarantee no details of a sentence are lost: After analyzing and storing a given sentence, an integrity system retrieves the sentence from the knowledge structure and compares it with the original sentence. Any loss of detail is considered to be a violation of the system's integrity.

Building history
By keeping track of changes – rather than overwriting knowledge – a history is build in the knowledge structure. Therefore, it is able to e.g. differentiate past tense from present tense.

3.3 Semantics

Semantics is the process of autonomously putting words (and word combinations) in the corresponding context. And a *context* is the distinction between ambiguous objects:

- The more specific and the more consistent the distinction of a word in a certain context, the clearer the meaning of that word in that context;
- The more fundamental the contexts of a given word, the deeper meaning of that word.

Semantics can be generated automatically by *associating* and *discriminating* knowledge, the first two capabilities from the definition of intelligence:

Associating or combining of knowledge
Knowledge can be associated or combined by implementing rules like:

- Classes can only be combined with related classes, which are classes of the same word type having the same subclasses[2] and/or the same relations[3];
- Subclasses can only be combined with related subclasses, which are subclasses of the same word type having the same classes and, if present, also the same relations;
- Relations can only be combined by related relations, which are relations of the same word type having the same classes and subclasses.

Discriminating or differentiating of knowledge
Knowledge can be discriminated or differentiated by implementing rules like:

- Subclasses will be differentiated when conjunction "*or*" is used to exclude them, like in definition sentence: "*A parent is a father or mother.*". So, when a particular father is named, this definition sentence excludes the possibility for that father of being a mother as well;
- When the sentence is given: "*Boston is a city is the USA as well as a city in the UK.*", the knowledge will be able to differentiate both occurrences of Boston;
- By keeping track of changes, the system will be able to differentiate past tense from present tense.

3.4 Static and dynamic structures

An article followed by a noun provides information whether that sentence is static or dynamic. A definition sentence is static and has an indefinite article. However, a dynamic structure has a definite article and can change over time, like in the situation "*Bush is **the** president of the USA.*" has changed when Obama was elected: "*Bush **was** the president of the USA.*" and "*Obama **is** the (current) president of the USA.*". So, a changed situation can influence other knowledge as well. Only the knowledge that is associated with the changed information, might change too. This mechanism is an important characteristic of natural language. And it is used extensively in providing programming in natural language. See: 3.11 Bridging the gap between natural and programming language.

Using mechanisms like this, makes it possible to implement techniques with a deeper level of "understanding" than traditional ontology, NLP and Question Answering techniques.

3.5 Reasoning in natural language

By implementing upper-ontology[4] techniques in natural language, the system is be able to reason in natural language: generating new knowledge, asking questions and detecting conflicts.

Domain ontology will be implemented as text files containing information about certain subjects, written in natural language.

2 classes and subclasses according the definitions of ontology
3 Relations are words that are preceded by a preposition, like in sentence: "*John is the father of Paul.*". In this case, "*Paul*" is the relation word, because Paul has a (child) relation with "*John*".
4 also known as top-level ontology or foundation ontology

Conclusions
An example of an automatically generated conclusion: Given the sentence "*John is the father of Paul.*" will result in: "*Paul has a father (called John).*".

Assumptions
Given: "*A person is a man or a woman.*" and "*John is a man.*". The conclusion "*John is a person.*" can not be justified. So, it is called an assumption, expressing the uncertainty.

Questions
Given: "*A parent is a father or a mother.*" and "*John is a parent.*". Without further knowledge about the gender of John, the system will conclude: "*John is a father or a mother.*". However, in order to encourage the user to complete the knowledge, the conclusion will be converted to a question: "*Is John a father or a mother?*".

Conflicts
Given: "*A parent is a father or a mother.*" and "*John is a father.*". When a sentence is entered like "*John is the mother of Paul.*", it will not be accepted, because it is in conflict with "*John is a father.*".

3.6 Writing grammatically correct sentences

Traditionally, sentences are parsed for reading. However, the grammar rules can also be used to parse the knowledge structure in order to write the knowledge in grammatically correct sentences.

3.7 Response

By comparing the new knowledge to the present knowledge, the software is able to respond to user about changes made in the knowledge structure. For example, when conclusions or assumptions are confirmed, when assumptions are invalidated, when the same, similar, related or more specific questions are asked, when questions are answered, or when conflicting knowledge is entered, the system will respond accordingly.

3.8 Semantic ambiguity

Two types of ambiguity can be distinguished: **static** ambiguity and **dynamic** (e.g. time-related) ambiguity. The difference is determined whether that sentence is static or dynamic.

An example of static ambiguity:
- "Boston is **a** city in both the United States and the United Kingdom".

An example of dynamic ambiguity:
- "Bush is inaugurated as **(the)** president of the United States.", because George H. W. Bush was inaugurated in 1989, and his son George W. Bush was inaugurated in 2001, and re-inaugurated in 2005.

3.9 Autonomous semantic disambiguation

When a sentence is entered and its context (semantics) is not clear, the system can either:

- use deduction to determine which context is meant by the user;
- make an assumption, when the meant context cannot be determined, but when it is quite obvious;
- or ask a question, when the system has no clue about the context.

3.10 Intelligent Answering of "is" Questions

Because the knowledge structure is build by *associating* and *discriminating* of knowledge, questions from the user can be answered in an intelligent way.

Examples:

- A user question like "*Is John the mother of Paul?*" can be answered human-like by "*No, John is the father of Paul.*" together with: "*Ann is the mother of Paul.*";
- A user question like "*Is Bill the father of Laura?*" can be answered human-like by "*I only know: Bill is a father.*", when it is unknown if Bill is the father of Laura;
- A user question like "*Is Joe a child of John?*" can be answered human-like by "*I am not sure, but I assume: Joe is probably a child of John.*", when the answer has a higher level of uncertainty then the question itself – in other words – when the answer is an assumption.

3.11 Bridging the gap between natural and programming language

By implementing grammar rules for accepting if-then-else sentences – like in programming languages – the system will be able to *execute* those sentences, which makes programming in natural language possible.

By implementing dedicated functions for specific imperatives, the system will be able to *execute* imperatives according a predefined functions.

Programming in natural language
By writing the playing rules of a game – like Connect-Four – as selection sentences in natural language, and by implementing a general game solving strategy – triggering by an imperative like the word "solve" – the system will be able to play the game against the user only by reading and executing the playing rules, written in natural language.

No programming is required to change the playing rules of the game: Just change the text and read the playing rules again.

4 Implementation

To be able to implement the described methods and techniques, the order of importance of the list below must be clear:

- foundation, most important;
- methods and techniques, less important;
- tools, least important.

Methods and techniques
Since the described methods and techniques have a fundamental approach, the most obvious methods and techniques – like NLP and ontology – might be insufficient to implement them.

Examples:

- Ontology is designed to represent a logical structure – rather than sentences in natural language;
- NLP techniques are designed to parse sentences – rather than to build a knowledge structure from them or to represent a logical structure;
- Both NLP and ontology are not designed to be compatible: formal language versus natural language, unlike the above described reasoning and knowledge representation techniques.

Tools
In the same way, the most obvious tools – like Prolog – might be insufficient to implement the described fundamental approach, because e.g. like ontology and Prolog are designed with logic in mind – ***pro**grammation en **log**ique* – rather than representing sentences in natural language.

Covering multiple disciplines
The system described above combines separate disciplines in the field of AI, in which the methods, techniques and tools are also designed to only cover one specific discipline. Therefore, it will be hard to combine those methods, techniques and tools, in order to implement the described system.

Contribution to science
The described system combines separate disciplines in the field of AI: Parsing (NLP), reasoning (ontology), semantics and writing new knowledge in grammatically correct sentences.

5 Conclusion

By going back to basics, defining a fundamental relation with intelligence and by using techniques described above, fundamental AI can be implemented through natural language.

