*i*Journals : International Journal of Social Relevance & Concern (IJSRC)
ISSN-2347-9698
Volume 10 Issue 9 September 2022

From the Least Effort Point of View, an Investigation on Language

Author: Weihan Huang

Master of Computer Science Department, State University of New York, at Buffalo, U.S.A.
Master of Physics Department, National Hsing Hua University, Taiwan
Email : weihanh@yahoo.com.tw

DOI: 10.26821/IJSRC.10.9.2022.100902

ABSTRACT

Naming : Language. What is language? And why do we need language? This paper tries to answer these two questions from the least effort point of view. The mechanism of the least effort way is firstly given to answer the question of what language is. Then I show 5 major types of using language in the least effort way to carry information : 1. Function Call which carries information of function definition, 2. Referential Use which carries information of properties and functions, 3. Interpretations which carry more than one interpretation, 4. Association of Memory for fast search, and 5. Abstract Conceptualization with reasoning within small segments. These are all reasons why we need language. Secondly, I state the mechanism of least effort way in the reverse direction : from large information to small segments, in contrast with the mechanism in the forward direction : from small segments to large information. Lastly, some important examples are provided in programming language, natural language, logic language, and the poetry.

Keywords**: **Association, Information, Language, Least Effort

1. THE MECHANISM OF THE LEAST EFFORT WAY TO CARRY INFORMATION

Language is the least effort way to carry information [1]. The mechanism is by using small segments to be associated to large information. Here are 5 major types of using language to be the least effort way to carry information.

1.1 Function call [2], carries information of function definition [2]

A function call, which is the function name with actual arguments [2], carries the information of function definition. A function name is about a sentence long, carries the function definition whose length could be from 10 to 30 lines of sentences roughly speaking.

For example of a merge sort [3], the code length is 30 to 40 lines. So the function call in one sentence carries about 35 times sentence information. This is a realization of least effort way by using small segments to be associated by large information.

And for the example of "sort an array", the function call only contain 3 segments, and it can sort an array with arbitrary length N, say containing 1000 elements. The complexity [4] of this example is

N*Log(N)~=3000. i.e. This information is counted for the loop length inside the function definition, so we can have a more exact way to express the complexity of information. Hence the 3 small segments "sort an array" carries the information of its complexity N*Log(N).

1.2 Referential Use [5], carries information of properties and functions

When I see a stone, it is so large and heavy that I am not able to carry it with me. However, by the referential use of natural language I can use "the stone" I see on the grass land, so that I can talk about the stone, its color : gray, its shape : large and round, and its position : on the grass land. And functions of the stone are that it can be sited, and it can be used to make a house. Therefore the referential use of "the stone", which are only 2 small segments, carrying the information of its properties : color, shape, position, and its functions : sitting, making a house. And the stone is what I cannot carry, but "the stone" is what I can carry in language. So it follows the mechanism of least effort that we use small segments to be associated to large information.

1.3 Interpretations [6], carry more than 1 interpretation

For example of a natural language "A boy watches the girl with a telescope.", it contains two interpretations,

A boy watches (the girl with a telescope).

A boy watches (the girl) with a telescope.

i.e. Two or more functions are applicable for one function name. It is an error of ambiguity for programming languages because they require definite semantics, however it is a zoom for imaginations of arts in which more interpretations are welcome. Thus small segments, the function name, are associated to more than one function definitions. And this follows the mechanism of using small segments to be associated to large information.

1.4 Association of Memory [7], fast search

Two words are associated if they appear together in a sentence or in a paragraph, and the strength of the association is the count of how often they appear together. Please note that association is one to many. For example of "memory", it can be associated to "association" and "fast" and "search" which are the title of this section. And "memory" could also be interpreted as the song "Memory" of the Broadway [8] musical [9] "Cats" [10]. And "memory" could also be linked to the memories of family and friends. Therefore it searches in a way of one-to-many, and hence the mechanism of using small segments to be associated to large information is realized.

Other examples of association of memory are like

1.4.1 talking : search for related words

1.4.2 reading : create interpretations

1.4.3 writing : search for similar concepts

1.5 Abstract Conceptualization [11] : reasoning within small segments

In algebra, variables are used to represent the actual world objects. Hence an abstract conceptualization is done for algebra. And then we can do reasoning within the small segments, here only one segment for variables x,y,z. For example, let x be the number of cows in a room, let y be the number of chickens in the room. And we know that there are totally 7 animals in the room, and totally 20 legs in the room. The question is to compute the number of cows and

chickens.

Totally 7 animals ==> abstraction ==> x+y=7

Totally 20 legs ==> abstraction ==> 4x+2y=20

Reasoning within small segments ==> solve the equations.

Then we get x=3, y=4 ==> come back to real life 3 cows, 4 chickens.

The same is for axiom proof system in logic, which I will address it in section 3.3.

2. A VIEW OF INFORMATION COMPRESSION [12]

2.1 Reverse the direction

If we reverse the direction of the least effort point of view of language, then it will look like that we are doing Information Compression. For example of the function call, the result of the compression of the function definition is its function name. So what we spread the compressed information is its function name, and then the receiver will decode it back to function definition.

i.e. In the network of communication, we firstly compress the data function definition to its function name, then pass the function call(function name + actual arguments) to the receiver. Lastly the receiver decompresses it to the function definition. Here we assume people in the network have common semantic commitment, which means the function definition is the same for the same function name. (Art is an exception, where it could result in more than 1 interpretation.)

2.2 Title of a book , Abstract of an article

The title of the book can be viewed as the compression of the book. And the title or abstract of an article can be viewed as the compression of the article. And these all follow the mechanism of least effort which uses small segments to be associated to large information.

2.3 Compression of a life

The philosophical question "what is the meaning of a life?" requires the compression of a whole life into a single sentence to answer to this question. This is an example of information compression. And I will give a possible answer to this question later in this paper.

3. OTHER EXAMPLES

3.1 Programming Language [13]

In programming language we can have finite control of infinite loop [14]. The mechanism is by running two threads concurrently, where the first thread is the control thread, and the second thread is an infinite loop work thread. The control thread is waiting for users to issue commands like : start the second thread, stop the second thread, continue the second thread, end the second thread, and quit all.

Therefore infinite loops which potentially contain infinite information can be controlled by finite commands which are all 4 words. This is the way of using small segments to be associated to large information.

3.2 Natural Language

There is one philosophical question expressed in natural language : "what is the meaning of a life?". This question is actually the compression of a whole life to one sentence! One of my own answers is that "cherish the accompanying of beloved people around

and be happy". And the answer of this question will guide me to make action decision and planning.

3.3 Logic language : axiomatic proof system [15]

Let proposition P represents "John is here", and let proposition Q represents "Mary is here". These two representations are just the "Abstract Conceptualization" mentioned in 1.5. And then we will do "reasoning within small segments".

Assume we know that

1. P or Q
2. not Q

Please note that (P or Q) is equivalent to (not Q -> P). So we have

3. not Q -> P

Therefore by 2 and 3, we get

4. P

Hence we have P, not Q. i.e. We have "John is here" and "Mary is not here". So just like algebra, we firstly transform natural language sentences to propositions containing only one segment of letter. Then we do "reasoning within small segments" to get the results P and not Q. This follows the rule of 1.5 Abstract Conceptualization.

3.4 Art : Poetry

The sentence of a poem "Leave me the pleasant memory of the cat" has 3 interpretations.

1. The song "memory" of the musical [9] "Cats" [10]
2. The memory of the cat I raised
3. The metaphor of "cat" could be "literature"

So it follows from 1.3 Interpretations and hence it has the mechanism of using small segments to be associated to large information.

We can also think from the view of information compression that each word or each sentence of the poem is a condensation of a musical, a movie, or a book. When reading the word and the sentence, “memory” will be associated to the whole musical, the whole movie, or the whole book by 1.2 Referential Use, 1.3 Interpretations and 1.4 Association of memory.

4. CONCLUSION

I have provided a view of what language is by the least effort way. The mechanism of the least effort way to carry information is by using small segments to be associated to large information. There are 5 major types of the mechanism : Function call, Referential Use, Interpretations, Association of Memory, and Abstract Conceptualization. These are all reasons why we need language. Particularly, we can see languages from the way of information compression. Therefore we can condense the whole musical, the whole movie, the whole book, even for the whole universe and a whole life into small segments by Referential Use, which greatly speed up our thinking and communication from the least effort point of view!

5. REFERENCES

[1] Information

https://en.wikipedia.org/wiki/Information

[2] Function call, Function definition, Actual argument

In the function definition f(x,y)=x+y. The f is called the function name, and x and y are called the formal parameters. A function call f(1,2) contains its function name f and the actual arguments 1and 2.

[3] Merge sort

https://www.educative.io/edpresso/how-to-implement-a-merge-sort-in-java

[4] Complexity

https://en.wikipedia.org/wiki/Computational_complexity

[5] Referential use

https://en.wikipedia.org/wiki/Reference

[6] Interpretations

https://en.wikipedia.org/wiki/Interpretation_(philosophy)

[7] Association of memory

https://en.wikipedia.org/wiki/Associative_memory_(psychology)

[8] Broadway

https://en.wikipedia.org/wiki/Broadway_theatre

[9] Musical

https://en.wikipedia.org/wiki/Musical_theatre

[10] Cats

https://en.wikipedia.org/wiki/Cats_(musical)

[11] Abstract Conceptualization

https://en.wikipedia.org/wiki/Algebra

https://en.wikipedia.org/wiki/Logic

[12] Information compression

https://en.wikipedia.org/wiki/Data_compression

[13] Programming language

https://en.wikipedia.org/wiki/Programming_language

[14] Infinite loop

https://en.wikipedia.org/wiki/Infinite_loop

[15] Axiomatic proof system

https://en.wikipedia.org/wiki/Axiomatic_system