An Analysis of Word Formation and Usage in Computer Science

Introduction

In recent years, computer technology has experienced explosive development, and the number of articles in related fields are increasing day by day. However, when we read computer science papers, we do not feel that there is too much unfamiliar, specialized vocabulary to hinder our reading. Is it possible that scientists have deliberately avoided the use and manufacture of unfamiliar words in the development of computer science? As a discipline close to real life and production, does computer science only use a large number of common words that belong to a certain range, rather than using a large number of seemingly more professional words? By studying the knowledge of word roots, we hope to analyze the words used in computer papers from the perspective of word roots, and then verify or disprove our conjecture. In this article, We analyze computer science papers from different fields as well as ordinary articles from everyday life, and asserte that computer science does not actually use too many extreme words, but prefers to use some common words in certain fields.

Literature review

In previous studies of terminology, it has been asserted that "science of terminology has developed from practical issues such as guidelines and recommendations in order to remedy communicational deficits passing phases of intensified theorisation". Specifically, in computer science, "IT terminology reflects the cutting-edge innovations in the field, providing a framework for discussing and understanding these developments. IT terminology is the backbone of effective communication within the technology sector and across various industries", which means that computer science terminology is the vehicle for a lot of communication. Therefore, according to scholars, these terms must "promote clarity, efficiency, and collaboration while ensuring that the IT field remains adaptable to rapid technological changes." That means our term is "Precise" and "well-defined". Thus, We can infer that overly complex, unintelligible vocabulary does not occur very often in computer science.

In "Computer Jargon Explained", the author even covers the main computer terminology in only 68 words, which meant that there were not many unfamiliar words in computer science at that time -- but unfortunately, this book was written in the last century. As far as we know, there is no good paper on the current state of computer science vocabulary.

Research purpose

The purpose of this experiment is to verify our previous claims about the characteristics of words used in computer science. That is, in order to adapt to the development of technology, the vocabulary used in computer science is mostly accurate and concise, and it often involves a large number of words in a special field, but these words are not uncommon.

Results and discussion

Results

Subjects

To test our conjecture, we neede to obtain the lexical distribution of papers in the field of computer science and the lexical distribution of everyday articles. So we find three papers from different fields of computer science and use ChatGpt to generate

several daily articles. Then, we cut them all down to about 27,000 characters. Next, we use technology to count the roots involved in these words, and use these statistics to replace the root distribution used in computer papers and everyday articles, respectively.

Having prepared the data, we can proceed to analyze them.

First, we'll compare these distributions to see if they're similar. If not, we can verify that computer science articles tend to use words that are specific to a particular field, as opposed to words that are commonly used in everyday fields.

Next, we will analyze the most common roots to see if they are rare roots. If not, we can conclude that most words in computer science are not special at all.

Finally, we pick out the specialized words involved in these roots, study the formation of these words, and see how they appear, so as to analyze whether the generation of computer words is complex. If their composition is not complicated, they are easy to understand and easy to communicate.

Method

For data, we select three papers from the algorithmic field, the engineering field and the artificial intelligence field of computer science. We then use automatic recognition technology to extract the English words used in these articles. We hope that the amount of data used will not be too different, so we limit the number of characters used to around 27,000, and the excess will be deleted by us.

Then we refer to the roots that we need to memorize in class to form a set of roots. In the experiment, we only check whether the prefix and suffix of each word are words in the root set, and if they are, we add the word to the list of words corresponding to the root. The reason for checking only prefixes and suffixes is that if a root is the same as something in the middle of a word, it's probably just a coincidence -- for example, the root "hal" appears in "challenges", the root "art" appears in "heartbeat", "ni" appears in "awakening" -- obviously these words have nothing to do with these roots. To do so would introduce a huge amount of error. However, this error can be greatly reduced if only prefixes and suffixes are detected. In this way, by counting the number of occurrences of each root, we get a general distribution of roots for all articles. Next we check whether these distributions are similar. This is simple, we just need to calculate the KL divergences between every two distributions. All we need to know is that the higher the KL divergences, the closer the two distributions are. In order to define what distributions are close, we randomly divide ChatGpt generated essays into two classes, and define the calculated KL divergences for these two classes of essays to represent the difference values of "close" essays.

Next, we focus on the few roots that occur the most in each distribution, and empirically judge whether these roots are rare or not. If most of these roots are common, then the words corresponding to the text are not uncommon.

Finally, we select the specialized words related to the most frequently occurring roots, analyze their formation, and use them as a representative to speculate whether most computer science words are complex constructions, and then demonstrate our previous points.

Result

First we will show the KL divergences of the root distribution corresponding to several articles. The string in the table represents the name and/or kind of the paper or article. It should be mentioned that the KL divergences of the two parts of the article generated by gpt after being randomly separated are 1.539277 and 0.945604, and each value in the table represents the KL divergence of the distribution above with respect to the distribution on the left.

	BitSense(algorithm)	**deTector(engineering)**	**cyclegan(AI)**	**gpt**	**gpt1**	**gpt2**
BitSense(algorithm)	0.000000	3.389979	3.729920	4.653855	3.741466	4.999882
deTector(engineering)	2.147132	0.000000	3.636089	4.639050	3.575310	5.171565

	BitSense(algorithm)	deTector(engineering)	cyclegan(AI)	gpt	gpt1	gpt2
cyclegan(AI)	3.053928	3.656885	0.000000	5.120916	4.298043	5.478428
gpt	4.297653	4.847775	4.507986	0.000000	0.403789	0.207736
gpt1	4.631582	5.399073	6.550042	2.455197	0.000000	3.236411
gpt2	5.310985	6.400097	5.584640	0.908626	2.053661	0.000000

Next we will show a few of the most common roots for each paper/article.

Finally, we show the specialized words that correspond to these roots and how they are formed. It is worth noting that we find that most of these terms are polysemous words whose meanings in technical terms are synchronously derived from their original meanings. We will explain how these polysemous words come to mean something in computers.

Roots/Prefix/Suffix	Specialized words	Is it a polysemous and common word	How did this meaning evolve	Word Formation
ment	segment	yes	It is formed by splitting a storage unit into several segments, so it is called segment.	Affixation: seg(root) + ment(suffix)
dis	distribution	no	---	Affixation: Dis(prefix) + tribut(root, to give) + ion(suffix)
---	traffic	yes	The transmission of data is similar to the flow of vehicles in actual traffic, so the data in transit is called traffic.	Conversion: 'trafic'(Old French) → traffic
memor	memory	yes	The hardware used to store data is similar to the memory system of the human brain, so the term memory is borrowed to refer to it.	Affixation: memor(root) + y
matr	matrix	no	---	Affixation: matr(root) + ix(suffix)
fail	failure	yes	The computer fails to complete the relevant goal and crashes, meaning the same as failure, so failure has become a computer term.	Affixation: fail(root) + ure(suffix)
loc	locate	yes	The process of finding a fault in the system is the process of locating the fault, so the word locate is used.	Affixation: loc(root) + ate(suffix)
dia	diagnose	yes	The computer error is similar to the patient's condition, so 'diagnose' is used to describe the process of finding a problem for the computer.	Compounding: Dia(root) + gnos(root)
monit	monitor	yes	The act of waiting for the value of a variable to change is monitoring that	Affixation: monit(root) + or(suffix)

Roots/Prefix/Suffix	Specialized words	Is it a polysemous and common word	How did this meaning evolve	Word Formation
			variable.	
imag	imagine	no	---	Affixation: Imag(root) + ine(suffix)
cycl	cyclegan	no	---	Acronymy: Cycle Generative Adversarial Network
photo	photograph	no	---	Compounding: photo(root) + graph(root)
gener	generator	yes	A program module used to output pictures is the module that generates pictures, so it is called a generator.	Affixation: gener(root) + ator(suffix)
fig	figure	no	---	Affixation: fig(root) + ure(suffix)
---	model	yes	A set of program fragments with specific functions is likened to a module on an industrial production line, so it is called a model.	Conversion: "modulus"(Latin) → model

Segment

Affixation: Seg (root) + -ment (suffix)
Distribution

Affixation: Dis (prefix) + Tribu (root) + -tion (suffix)
Measurement

Affixation: Measur (root) + -ment (suffix)
Traffic

Conversion: Originally from "trafic" in Old French, converted into English without affixes.
Memory

Affixation: Mem (root) + -ory (suffix)
Matrix

Affixation: Matr (root) + -ix (suffix)
Failure

Affixation: Fail (root) + -ure (suffix)
Locate

Affixation: Loc (root) + -ate (suffix)
Diagnose

Affixation: Dia (prefix) + Gnos (root) + -e (suffix)
Monitor

Affixation: Moni (root) + -tor (suffix)
Imagine

Affixation: Imag (root) + -ine (suffix)
CycleGAN

Acronymy: Cycle Generative Adversarial Network
Photograph

Compounding: Photo (root) + Graph (root)
Generator

Affixation: Gener (root) + -ator (suffix)
Figures

Pluralization: Figure (root) + -s (plural suffix)
Model

Conversion: Originally from Latin "modulus," converted into English without affixes.

表格，图表，不要推导公式，不写统计过程，统计术语做页下脚注

Discussion

（小标题）

（小标题）

结合图表，实验结果，文献理论讨论分析： why， how

Conclusion

References
Appendix