[41]

Coleman, R. (2018). Aesthetics Versus Readability of Source Code. International Journal of Advanced Computer Science and Applications, 9(9), 12–18. <https://doi.org/10.14569/ijacsa.2018.090902>

**KEYWORDS AUTOR:**

Programming style; fractal geometry; readability

**TÍTULO**:

Estética versus legibilidad del código fuente

**DATASET**:

<https://github.com/roncoleman125/Pretty>

**LENGUAJE DE PROGRAMACIÓN:**

C

análisis de imagen

**TAGS**

LEGIBILIDAD; ESTETICA; ESTILOS DE PROGRAMACION; ESTILOS DE EMBELLECIMIENTO; MEJORES PRACTICAS; EVOLUCION; METRICAS;

**PARA ANEXAR A DOCUMENTO:**

**\*\* INICIO \*\*\***

**CONCEPTO LEGIBILIDAD**

Algunas personas pueden confundir el concepto de legibilidad con la estética de un código fuente. La legibilidad hace referencia a la comprensión del código, mientras la estética, aesthetics en inglés, hace referencia a apreciar el código como una obra de arte, es decir, sin intentar "leer" o "comprender" su función. El término de "Embellecer" el código hace referencia a mejorar su estilo de acuerdo a lo que se considera bien escrito en términos de diseño y estructura. Para "volver feo" el código se hace lo contrario, e incluso hay herramientas, conocidas como "ofuscadores", que utilizan técnicas anti-estilo generalmente para fines de seguridad de la información. El modelo de belleza se encuentra en las guías de estilo, los estándares de codificación, políticas de codificación organizacional, libros, reportes de investigación, códigos de ejemplo,blogs, etc.

Coleman y Gandhi (2015) identificaron los “principios básicos” de un buen estilo de programación, los cuales son: 1) usar el espacio en blanco con prudencia; 2) elegir nombres mnemotécnicos; e 3) incluir documentación.

Se considera que al aplicar un buen estilo de programación y hacer que el factor de belleza aumenta en un código fuente, éste tiende a ser más legible, pero no habían estudios que lo comprobaran. Por ello, Coleman (2018) realizó un estudio para evaluar la relación entre los factores de belleza y los modelos de legibilidad del código fuente, analizando 11 estilos de belleza y antibelleza del código con 8 métricas de legibilidad, encontrando como conclusión que no están tan correlacionados como se creía y que la única relación fuerte se encuentra asociada con los estilos del principio 1 del uso de los espacios en blanco.

**MODELOS DE LEGIBILIDAD**

|  |  |  |
| --- | --- | --- |
| **Autor** | **Métricas** | **Categoría** |
| Halstead | Longitud del programa (N), vocabulario del programa (n), volumen del programa (V), dificultad (Df), esfuerzo (E) | Halstead |
| Buse y Weimer (BW) | Aprendizaje supervisado para entrenar un clasificador bayesiano para asociar las medidas de Halstead con los juicios humanos en un fragmento de código. | Aprendizaje de máquina |
| Posnet, Hindle, and Devanbu (PHD) | mejoraron el de Buse usando ingeniera de software clásica y teoría de la información. Aprendizaje fue derivado del análisis de regresión. | Aprendizaje de máquina |
| Abbas, Borestler, Cspersend, and Nordstrin (2009) | Modelo de fácil puntaje de legibilidad de software o SRES.  - métrica de longitud promedio de sentencia (ASL), que es el número promedio de tokens por declaración.  - métrica de longitud promedio de palabra (AWL)  SRES = ASL - 0.1 AWL  A mayor SRES, más dificil para leer. | Métricas inspiradas en la prosa |

**APLICACIONES**: Aplicación de estilos de embellecimiento para mejorar la legibilidad del software.

**RETOS**: Evaluar la relación de los estilos de embellecimiento del código fuente con la legibilidad del mismo en varios lenguajes de programación.

**\*\* FIN \*\***

**RESUMEN COMPLETO:**

Relación entre el estilo del programa (11 factor beauty) y la legibilidad (8 modelos). Cuando el factor de belleza aumenta, tiende a ser más legible

LENGUAJE DE PROGRAMACIÓN:

C

análisis de imagen

TRABAJOS RELACIONADOS A LA RELACIÓN ENTRE BELLEZA Y LA LEGIBILIDAD:

|  |  |  |  |
| --- | --- | --- | --- |
| Año | Autor | Factor beautifier | Descripción |
| 1979 | AT&T Bell Labs | Algoritmo cb (C Beautifier) | Cambia el formato acorde a las reglas del estilo K&R |
|  | Kokol et al. |  | Correlación entre caracteres y los token. |
| 2015 | Coleman y Gandhi |  | Propusieron un modelo fractal relativista y mostraron que la belleza estaba correlacionada de manera débil a moderada con la complejidad del software en direcciones que coincidían con las recomendaciones de estilo.  Identificaron los “principios básicos” de un buen estilo de programación: 1) usar el espacio en blanco con prudencia; 2) elegir nombres mnemotécnicos; y 3) incluir documentación. |
| 2017 | Coleman y Boldt |  | Investigaron el desorden que a menudo se introduce en el código a través del mantenimiento y mostró que la belleza tenía una correlación débil o moderada con la entropía. |
| 2017 | Coleman and Rahtelli |  |  |

CONCEPTOS:

**Legibilidad**: hace referencia a la comprensión del código.

**Estética (Aesthetics)**: hace referencia a la apreciación del código, como un arte.

"Embellecer" el código es mejorar su estilo de acuerdo a lo que se considera bien escrito en términos de diseño y estructura. Para "volver feo" el código se hace lo contrario; de hecho, existen herramientas, conocidas como "ofuscadores" que utilizan técnicas anti-estilo generalmente para fines de seguridad de la información. apreciar la forma del código como una obra de arte, es decir, sin intentar "leer" o "comprender" su función.

El modelo de belleza se encuentra en las guías de estilo, los estándares de codificación, políticas de codificación organizacional, libros, reportes de investigación, códigos de ejemplo,blogs, etc.

Transformaciones que preservan la semántica:

|  |  |  |  |
| --- | --- | --- | --- |
| **Objetivo** | **Estilo** | **Principio** | **Procedimiento** |
| Embellecimiento | GNU | 1 | Aplicar el estilo |
| Embellecimiento | K&R | 1 | Aplicar el estilo |
| Embellecimiento | BSD | 1 | Aplicar el estilo |
| Embellecimiento | LIN | 1 | Aplicar el estilo |
| Embellecimiento | MNE | 2 | Refactorizar nombre a nemotécnico |
| Embellecimiento | REC | 3 | Adicionar comentarios |
| Anti Embellecimiento | NOI | 1 | Remover identación |
| Anti Embellecimiento | R2 | 1 | Identación aleatoria con 1 o 2 espacios |
| Anti Embellecimiento | R5 | 1 | Identación aleatoria con 1 o 5 espacios |
| Anti Embellecimiento | NON | 2 | Refactorizar nombres para ser menos nemotécnicos. |
| Anti Embellecimiento | DEC | 3 | Remover todos los comentarios |

TRABAJOS RELACIONADOS LEGIBILIDAD

|  |  |  |
| --- | --- | --- |
| Autor | Métricas | Categoría |
| Halstead | Longitud del programa (N), vocabulario del programa (n), volumen del programa (V), dificultad (Df), esfuerzo (E) | Halstead |
| Buse y Weimer (BW) | Aprendizaje supervisado para entrenar un clasificador bayesiano para asociar las medidas de Halstead con los juicios humanos en un fragmento de código. | Aprendizaje de máquina |
| Posnet, Hindle, and Devanbu (PHD) | mejoraron el de Buse usando ingeniera de software clásica y teoría de la información. Aprendizaje fue derivado del análisis de regresión. | Aprendizaje de máquina |
| Abbas, Borestler, Cspersend, and Nordstrin (2009) | Modelo de fácil puntaje de legibilidad de software o SRES.  - métrica de longitud promedio de sentencia (ASL), que es el número promedio de tokens por declaración.  - métrica de longitud promedio de palabra (AWL)  SRES = ASL - 0.1 AWL  A mayor SRES, más dificil para leer. | Métricas inspiradas en la prosa |

RESULTADOS

Evaluados con el coeficiente de correlación Spearman

|  |  |
| --- | --- |
| **Relación** | **Conclusiones** |
| Belleza vs métricas de Halstead | Mejores correlaciones para anti embellecer:  NOI = -0.55, NON = 0.48  Mejores correlaciones para embellecer:  GNU = -0.58, LIN = -0.02, BSD = -0.43, K&R = -0.46 |
| Belleza vs métricas de aprendizaje de máquina | Mejores correlaciones para anti embellecer:  NOI = 0.34, NON = -0.36  Mejores correlaciones para embellecer:  GNU = 0.37, K&R = 0.31, LIN = 0.34 |
| Belleza vs métrica inspirada en la prosa | Mejores correlaciones para anti embellecer:  NOI = 0.38  Mejores correlaciones para embellecer:  GNU= -0.44, K&R = -0.35, BSD = -0.41 |
| Conclusiones finales | NOI es un indicador contrario  La legibilidad mejora con el principio #1 del espacio en blanco.  GNU es el estilo más legible |

**\*\* FIN \*\*\***