[56]

Xu, W., Xu, D., & Deng, L. (2017). Measurement of Source Code Readability Using Word Concreteness and Memory Retention of Variable Names. In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC) (pp. 33–38). IEEE. <https://doi.org/10.1109/COMPSAC.2017.166>

**KEYWORDS AUTOR:**

Code Readability; Variable Definitions and References; Word Concreteness; Memory Retention;

**TÍTULO**:

Medición de la legibilidad del código fuente utilizando la concreción de palabras y la retención de memoria de nombres de variables

**PARA ANEXAR A DOCUMENTO:**

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

**DATASET**:

Fragmentos: proyectos de código abierto obtenidos de SourceForge y GitHub.

**DOCUMENTACIÓN:**

Lista completa de abreviaciones:

<https://raw.githubusercontent.com/Bowie-State-University/CS-Courses/master/research/UnigramAnalysisDataPaper/topAbbWordPair.txt>

Lista de unigrams:

<https://raw.githubusercontent.com/Bowie-State-University/CS-Courses/master/research/variables/varNameAll/unigram/unigramAll.txt.txt>

Variables extraídas:

<https://github.com/Bowie-State-University/ReadablilityCaculate/tree/master/inputs>

**HERRAMIENTAS**:

Para segmentación de palabras en Inglés:

<https://pypi.org/project/wordsegment/>

**LENGUAJE DE PROGRAMACIÓN:**

Java

**TAGS**

LEGIBILIDAD; MODELO; WCMR; METRICAS; CONCRECION DE PALABRAS; RETENCION DE MEMORIA; NOMBRES DE VARIABLE;

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Descripción** | **Resultados** |
| Xu et al (2017) | No se basó en el enfoque de características, sino en el enfoque basado en WCRM: concreción de palabras y retención de memoria de nombres de variable.  El modelo midió la legibilidad del código fuente directamentes desde los nombres de variable sin un proceso de entrenamiento.  **MÉTRICAS:**  **Legibilidad de la variable:**  La legibilidad de una variable “v” en un método “m”, legibilidad(v, m), está definida como:  Legibilidad(v,m) = Concreción(v) X Retención(v)  **Legibilidad de un método:**  La legibilidad de un método o constructor “m” en una clase “c” está definida como la legibilidad promedio de todas las variables “V” usadas en “m”, incluyendo variables locales, parámetros, constantes nombradas, variables instanciadas y variables de clase.  Legibilidad(m,c) = (sumatoria(legibilidad(vi,m))) / |V|  donde,  i empieza en 1 y finaliza en n  **Legibilidad de la clase:**  La legibilidad de la clase C en un programa orientado a objetos P es definido como la legibilidad promedio de todos los métodos y constructores en “C”.  Legibilidad(m,P) = (sumatoria(legibilidad(mi,c))) / |M|  donde,  i empieza en 1 y finaliza en n  **Legibilidad del programa:**  La legibilidad de un programa orientado a objetos P está definida como la legibilidad promedio de todas las clases en P.  Legibilidad(P) = (sumatoria(R(ci,P))) / |C|  donde,  i empieza en 1 y finaliza en n  MÉTODOS  Regresión lineal para analizar correlación entre la legibilidad del código y otras variables dependientes. Se usó R2, el coeficiente de determinación, para medir la bondad de ajuste del modelo de regresión lineal. | La legibilidad del código fuente basada en WCMR está fuertemente correlacionada con la tasa de alertas de defectos. Las alertas de defectos con mayor correlación son las clasificadas en: malas prácticas, vulnerabilidad de código malicioso y advertencias de corrección. |

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

**RESUMEN COMPLETO**

Xu et al (2017) presentaron un enfoque para medir automáticamente la legibilidad del código fuente basado en la concreción de palabras y retención de memoria de nombres de variable, conocido como WCMR por sus siglas en inglés. Los autores sustentaron que la legibilidad de cada variable está determinada por qué tan fácil su significado es memorizado (concreción de palabras) y que tan retenido (retención en memoria) es su significado por los desarrolladores cuando la variable es referenciada. El modelo permite medir la legibilidad del código fuente directamentes desde los nombres de variable sin un proceso de entrenamiento.

La novedad del enfoque está en los siguientes 3 aspectos:

1. Los nombres de variables en el código fuente no siempre son simples palabras de diccionario, así que el enfoque mide el valor de la concreción de nombres comunes que pueden tener abreviaciones, prefijos, sufijos y múltiples palabras (nombres multigram). Para un nombre multigram, se exploran la similitud conceptual y la similitud relacional entre las palabras unigram en el nombre multigram para medir el valor de concreción general.
2. Utiliza las distancias de lectura de las variables y la densidad del uso variable para cuantificar la disminución y la mejora de la retención de memoria, respectivamente.
3. Proporciona evaluaciones formativas y sumativas que pueden ayudar a los programadores a mejorar la legibilidad del código localmente (declaración y método) y globalmente (clase y proyecto).

DEFINICIONES DE WCMR

CONCRECIÓN

La concreción evalúa el grado en que el concepto denotado por una palabra se refiere a una entidad perceptible. Una palabra concreta es más fácil de aprender, recordar y procesar que una palabra abstracta.

Los nombres de variable son clasificados en 2 categorías:

* Unigram: una simple palabra. El valor de concreción de una palabra “w”, que se denota como fc(w), está determinado por su valor en el diccionario de forma directa o indirecta. Si no se encuentra el valor es cero.
* Multigram: es un nombre compuesto, es decir, con varios unigram. Su valor de concreción es calculado basado en la similaridad de palabras de las que está compuesto, cuyo rango va de 0 a 1, donde 0 es que las palabras son diferentes conceptualmente o no relacionadas y 1 que son sinónimos. Hay 2 tipos de similaridad: similaridad conceptual y similaridad relacional. Usaron la herramienta de análisis semántico latente (LSA por sus siglas en inglés) para calcular la similaridad de palabras.

|  |  |  |
| --- | --- | --- |
|  | Bajo grado - 0 | Alto grado - 1 |
| Similaridad conceptual | Conceptualmente diferentes | Sinónimos |
| Similaridad relacional | Nunca o rara vez coinciden en una sentencia. | Siempre coinciden en pares en una sentencia en una posición particular. |

RETENCIÓN DE MEMORIA DE NOMBRES DE VARIABLE

La retención de memoria mide la habilidad de la mente humana para retener información. Los factores para determinarla son: 1) la disminución de la retención de memoria que sucede con el tiempo; 2) la mejora de la retención de memoria, que se logra usando la variable de forma adicional y se mide por la densidad de uso de la variable.

**LEGIBILIDAD DEL CÓDIGO FUENTE**

**Legibilidad de la variable:**

La legibilidad de una variable “v” en un método “m”, legibilidad(v, m), está definida como:

Legibilidad(v,m) = Concreción(v) X Retención(v)

**Legibilidad de un método:**

La legibilidad de un método o constructor “m” en una clase “c” está definida como la legibilidad promedio de todas las variables “V” usadas en “m”, incluyendo variables locales, parámetros, constantes nombradas, variables instanciadas y variables de clase.

Legibilidad(m,c) = (sumatoria(legibilidad(vi,m))) / |V|

donde,

i empieza en 1 y finaliza en n

**Legibilidad de la clase:**

La legibilidad de la clase C en un programa orientado a objetos P es definido como la legibilidad promedio de todos los métodos y constructores en “C”.

Legibilidad(m,P) = (sumatoria(legibilidad(mi,c))) / |M|

donde,

i empieza en 1 y finaliza en n

**Legibilidad del programa:**

La legibilidad de un programa orientado a objetos P está definida como la legibilidad promedio de todas las clases en P.

Legibilidad(P) = (sumatoria(R(ci,P))) / |C|

donde,

i empieza en 1 y finaliza en n

EXPERIMENTO

métodos: regresión lineal para analizar correlación entre la legibilidad del código y otras variables dependientes. Se usó R2, el coeficiente de determinación, para medir la bondad de ajuste del modelo de regresión lineal.

RESULTADO

La legibilidad del código fuente basada en WCMR está fuertemente correlacionada con la tasa de alertas de defectos. Las alertas de defecto con mayor correlación son las clasificadas en: malas prácticas, vulnerabilidad de código malicioso y advertencias de corrección.