Análisis Automático de Indicadores de Calidad de Historias de Usuario mediante Medidas Difusas

Carlos Casanova[1,2], Karina Cedaro[1], Rossana Sosa Zitto[1]

[1] Grupo de Investigación en Ingeniería de Software (GIISW)
Universidad Autónoma de Entre Ríos, Facultad de Ciencia y Tecnología
[2] Grupo de Investigación sobre Inteligencia Computacional e Ingeniería de Software (GIICIS)
Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay
casanovac@frcu.utn.edu.ar, cedaro.karina@uader.edu.ar, sosa.rossana@uader.edu.ar

Resumen

La calidad de los entregables de la Elicitación de Requisitos es causa de gran parte de los desvíos significativos en los proyectos de desarrollo de software por el alto impacto de sus consecuencias sobre el producto final. Aun así las empresas no invierten los recursos suficientes, cuando ello les permitiría reducir esfuerzos, costos y obtener ventajas en un mercado altamente competitivo. Si bien la Ingeniería de Requisitos intenta con sus aportes mejorar la situación, el alcance de sus propuestas no siempre aplica a las PyMEs desarrolladoras de nuestra región, industrias que se enfrentan a diario con falta de recursos, habilidades y experiencia en su búsqueda por crear software de calidad y sobrevivir en el mercado. En este marco el presente trabajo propone una herramienta basada en la aplicación de técnicas de análisis de textos que permite complementar el rol de los ingenieros de requisitos en lo que hace a la verificación de los resultados de este proceso. Tal herramienta se sustenta en un modelo basado en medidas difusas que determina indicadores de calidad en conjuntos de historias de usuario en tres características clave: no ambigüedad, ausencia de conflictos y unicidad. La herramienta se utiliza en conjuntos de historias de usuario recopiladas por los autores para identificar fortalezas y debilidades, obteniendo resultados prometedores que animan a continuar con la investigación.

***Palabras clave**: Ingeniería de Requisitos, Análisis Automático de Texto, Lógica Difusa, Historias de Usuario.*

1. Introducción

Actualmente, la industria nacional del sector SSI (Software y Servicios Informáticos) tiene como unos de sus retos fundamentales implementar aplicaciones que sean entregadas a tiempo, que no involucren presupuestos elevados y que satisfagan las necesidades del usuario, utilizando para ello metodologías y herramientas que guíen el proceso de desarrollo de Software [1]. Tal proceso es la aplicación de un conjunto de actividades, acciones y tareas que se ejecutan para crear algún producto de trabajo. Se obtiene un producto de alta calidad si estas actividades, acciones y tareas, se combinan con procedimientos y técnicas de la ingeniería de software [2]. Por ello otra exigencia paralela que ocupa a la industria del software es la calidad de los recursos usados en sus procesos, debido a la directa implicancia sobre los resultados. Este tema día a día genera mayor interés, como lo demuestra el informe presentado en 2019 por la OPSSI (Observatorio Permanente de la Industria del Software y Servicios Informáticos) en el cual el 42% de las firmas realizaron mejoras en la calidad en el proceso de desarrollo [3].

Existen diversos estudios respecto de los fracasos en la industria del software. Tal es el caso del reporte del estudio presentado por la consultora internacional Standish Group, en el que se cita que tan solo el 32% de los proyectos de desarrollo de software se pueden considerar exitosos; el 44% se entregaron fuera de plazo, excedieron su presupuesto y no cubrieron la totalidad de las características y funcionalidad pactada; y el 24% de los proyectos fueron cancelados. En este

mismo estudio se puntualiza que el principal factor para el fracaso de un proyecto de desarrollo de software radica en la mala calidad de los requisitos y la definición poco clara de los mismos [4].

Por definición la Ingeniería de Requisitos (IR) es la disciplina que tiene por objeto el análisis, documentación y validación de las necesidades y/o requerimientos de las partes interesadas para el desarrollo de un sistema [5]. Su objetivo es desarrollar una especificación completa, consistente y no ambigua de los requisitos, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema. Sin embargo, los requisitos se describen mediante el uso de texto, que está orientado al cliente y no es apropiado para el desarrollador. Las descripciones ambiguas y los requisitos fragmentados en varios artefactos también socavan la comprensión. Los desarrolladores señalan que se requiere mucho esfuerzo para aclarar dudas cuando la especificación de requerimientos es insuficiente o se enfoca solo en los requisitos funcionales aumentando el esfuerzo de codificación, prueba y mantenimiento [6].

En este sentido, una de las notaciones para la especificación de requisitos que se ha popularizado desde hace ya algunos años es la Historia de Usuario, sobre todo con la adopción de las metodologías ágiles de desarrollo [7]. Si bien pueden encontrarse algunas diferencias, todos los autores reconocen los mismos tres componentes básicos de una historia de usuario: (1) un texto corto que describe y representa la historia de usuario, (2) conversaciones entre los stakeholders para intercambiar perspectivas sobre la historia de usuario, y (3) los criterios de aceptación. El texto corto que representa la historia de usuario captura los elementos esenciales del requisito: para quién es, qué se espera del sistema, y, opcionalmente, por qué es importante. El formato más difundido y el estándar *de facto* es: “Como <rol>, Quiero <medio>, [Para <fin>]. Por ejemplo: “COMO comprador QUIERO ver el listado de opciones de pago PARA escoger la que más me interese”.

En este trabajo se propone una herramienta basada en la aplicación de técnicas de análisis de textos que permite complementar el rol de los ingenieros de requisitos en lo que hace a la verificación de los resultados de este proceso. Tal herramienta se sustenta en un modelo basado en medidas difusas que determina indicadores de calidad en conjuntos de historias de usuario en características de calidad clave. En la sección 2 se describen criterios de calidad propuestos en el estado del arte en diversos frameworks, haciendo especial énfasis en QUS. La sección 3 presenta fundamentos del análisis de textos y cómo éste puede ser de utilidad en el contexto de IR. A continuación, en la sección 4, se presentan nociones básicas de lógica y medidas difusas, haciendo particular énfasis en la noción de relaciones de equivalencia difusas. Luego, en la sección 5 se describe el modelo propuesto para el análisis de historias de usuario. La sección 6 describe los experimentos realizados para una validación técnica inicial. Finalmente, la sección 7 describe las conclusiones del estudio y los trabajos futuros.

2. Criterios de calidad de Historias de Usuario

A pesar de su popularidad, el número de métodos para evaluar y mejorar la calidad de las historias de usuario es reducido. Muchos de los enfoques existentes, o bien emplean métricas altamente cualitativas (como las seis heurísticas mnemotécnicas INVEST [8] (Independiente-Negociable-Valorable-Estimable-Escalable-Testeable), o bien son guías genéricas para la calidad en entornos de desarrollo ágil. Un ejemplo de este tipo de guías es la *IEEE Recommended Practice for Software Requirements Specifications* [9], que define requisitos de calidad basados en ocho características: correcta, no ambigua, completa, consistente, ordenada por importancia/estabilidad, verificable, modificable, y trazable. El estándar, sin embargo, es demasiado genérico y se sabe que las especificaciones raramente cumplen con todos los criterios [10]. Otro ejemplo, ya dentro del ambiente ágil, es el *Agile Requirements Verification Framework* [11], que define tres criterios de verificación de alto nivel: completitud, uniformidad, consistencia y correctitud. Sin embargo, los criterios con frecuencia requieren para ser evaluados de información suplementaria y no estructurada adicional, la cual no está incluida en la historia de usuario primaria.

Uno de los marcos de trabajo que ha realizado un avance significativo en la evaluación y mejora de la calidad en historias de usuario es el denominado *Quality User Story* (QUS) *Framework* [7]. En él se proponen 13 criterios para determinar la calidad en historias de usuario en términos de sintaxis, semántica y pragmática. Los criterios sintácticos son: Bien formada, Atómica y Minimal; los semánticos: Conceptualmente sólida, Orientada al problema, No ambigua, y Ausencia de conflictos; y las pragmáticas: Oración completa, Estimable, Única, Uniforme, Independiente y Completa.

Cada uno de estos criterios está claramente definido y el *framework* cuenta con una herramienta, *Automated Quality User Story Artisan* (AQUSA) que soporta la actividad de ingeniería de requisitos cuando se usa este tipo de notación. El objetivo original de QUS fue aprovechar los avances de procesamiento del lenguaje natural (NLP) para obtener la Condición de Exhaustividad Perfecta (*Perfect Recall Condition*). Esta regla enfatiza la criticidad de alcanzar el 100% de exhaustividad en la identificación de defectos de calidad, sacrificando precisión si es necesario. En este contexto, la exhaustividad es el cociente entre la cantidad de defectos reales correctamente detectados y la cantidad de defectos reales totales (aún los no detectados), mientras que la precisión es el cociente entre los defectos reales correctamente detectados y todos los defectos detectados. Esta preferencia por los "falsos positivos" sobre los "falsos negativos" se debe a que si al analista se le garantiza que la herramienta no ha pasado por alto ningún defecto, entonces ya no necesita volver a verificar manualmente la calidad de todos los requisitos. Por supuesto que lograr el 100% de exhaustividad mediante herramientas automáticas es, si no imposible, al menos improbable [7].

La aplicación de QUS y AQUSA ha arrojado resultados prometedores, sobre todo en lo referente a los criterios sintácticos, pero los autores reconocen que aún debe estudiarse cómo y en qué medida incorporar los criterios semánticos y pragmáticos teniendo en cuenta el objetivo exhaustividad/precisión.

Finalmente, existen otras herramientas que utilizan la métrica de exactitud (*accuracy*, clasificaciones correctas sobre total de objetos), como QuARS, Dowser, Poirot y RAI, que intentan maximizar la precisión y tienen como objetivo entender los contenidos de los requisitos. Sin embargo, esto es todavía prácticamente imposible a menos que se produzca un gran cambio en los modelos de procesamiento del lenguaje natural [12].

Por esta razón, en este trabajo se proponen métricas que pueden incorporarse en una herramienta de análisis automático para abordar tres criterios de calidad presentes en QUS: no ambigüedad, ausencia de conflictos y unicidad. En la sección siguiente se definen estos criterios.

I. No ambigüedad

La ambigüedad es intrínseca en los requisitos expresados en lenguaje natural, pero el ingeniero que escribe historias de usuario debe evitarla todo lo posible. No sólo deben ser no ambiguas internamente, sino que deberían ser claras en relación a todas las demás historias de usuario. En [13] puede encontrarse una clasificación sobre los tipos de ambigüedad que pueden encontrarse en una especificación de requisitos.

II. Ausencia de conflictos y unicidad

Estos dos criterios están íntimamente relacionados. Para detectar este tipo de relaciones cada parte de la historia de usuario debe ser comparada con las de las demás usando una combinación de medidas de similitud que pueden ser sintácticas (por caso, distancia de Levenshtein) o semánticas (por ejemplo, mediante una ontología para determinar sinónimos, o mediante medidas de similitud contextual, como en este trabajo). Cuando se supera un umbral de similitud se requiere que un analista real examine las historias de usuario por el potencial conflicto o duplicación.

QUS plantea distintas medidas para estos dos criterios:

Medidas que afectan la **unicidad**:

Duplicación total: una historia de usuario HU1 es un duplicado total de otra HU2 cuando las historias son idénticas.

Duplicación semántica: una historia de usuario HU1 es un duplicado semántico de otra HU2 cuando las historias requieren lo mismo pero utilizando un texto distinto.

Medidas sobre posibles **conflictos**:

Diferentes medios, mismo fin: dos o más HUs que tienen el mismo fin, pero que se alcanza por distintos medios. Esta relación puede indicar: 1) una variación de una característica que debería ser explícitamente especificada en la historia para mantener un conjunto de historias no ambiguo; o 2) un conflicto sobre cómo se alcanza tal fin, por lo que una de las historias de usuario debería ser descartada para asegurar que las historias de usuario estén libres de conflictos.

Mismo medio, diferente fin: dos historias de usuario que utilizan el mismo medio para alcanzar diferente fin. Esta relación puede indicar que, si las HUs no poseen conflictos, entonces deberían ser combinadas en una misma historia de usuario; de lo contrario, constituyen múltiples puntos de vista que deben ser resueltos.

3. Análisis de Texto

La aplicación de técnicas de procesamiento y análisis de lenguaje natural se encuentran actualmente en auge. Una hipótesis fundamental de este trabajo es que pueden utilizarse este tipo de técnicas para el análisis de documentos de requerimientos, especialmente colecciones de historias de usuario, con el fin de establecer criterios de calidad (o ausencia de esta), como ambigüedad, conflictos intra documentales, duplicación y solapamiento, plausibilidad de la estimación, entre otros [14], [15]. Este análisis es capaz de evidenciar falencias en los documentos de requerimientos y constituir un insumo para la posterior recomendación de acciones tendientes a la mejora, por caso, la aplicación de técnicas específicas de elicitación. De ellas pueden beneficiarse ciertos sectores del mercado como las pequeñas empresas que no poseen un área de gestión de requisitos, o que son escépticos respecto de los beneficios que puedan aportar enfoques más sintácticos en el análisis de los mismos.

Muchas aplicaciones de análisis de texto utilizan el modelo vectorial con la técnica TF-IDF [16] (*term frequency-inverse document frequency*) por su simplicidad y rapidez. Sin embargo, existen algunos problemas con esta técnica. Por caso, TF.IDF no puede reflejar similitud entre palabras y sólo cuenta el número de palabras en común entre dos documentos, ignorando sinónimos y otra información sintáctica, como la división en rol, medio y fin de una historia de usuario. Además, cada inserción y eliminación de documentos implica el recálculo del valor de IDF, lo cual acarrea algunos problemas en datasets dinámicos.

En este trabajo, por lo tanto, se propone la utilización de conjuntos difusos como formalismo de base para la medición de similitud de palabras y frases. Las palabras se consideran similares si aparecen en contextos similares. Sin embargo, estas palabras similares no tienen por qué ser sinónimas ni pertenecer a la misma categoría léxica. Esta diferenciación es importante en el contexto de aplicación, ya que la utilización de enfoques que hacen uso de corpus externos para medir similitud entre palabras pueden resultar inadecuados debido a las diferencias de interpretación que aparecen en dominios específicos. Además, es bien sabida la estrecha relación entre la ambigüedad lingüistica y los enfoques difusos, como el CWW (C*omputing With Words)* [17].

4. Lógica y Medidas Difusas

La lógica difusa provee un modelo para realizar razonamiento "aproximado", en contraste con la lógica bivaluada clásica, donde los sistemas formales proveen modelos de razonamiento exacto donde no se permite ambigüedad. La Lógica Difusa está basada en el hecho de que la percepción humana involucra conjuntos "difusos" o "borrosos", esto es, clases de objetos en los cuales la transición entre pertenencia y no pertenencia es gradual en lugar de abrupta.

Sea X un conjunto clásico llamado universo del discurso. La pertenencia en otro conjunto clásico $A \subseteq X$ puede verse como una función característica μ_A de X a {0, 1}:

$$\mu_A(x) = \begin{cases} 1 & si \quad x \in A \\ 0 & si \quad x \notin A \end{cases} \qquad (1)$$

{0, 1} es llamado conjunto de valuación.

Si el conjunto de valuación puede ser el intervalo real [0, 1], A es llamado conjunto difuso. μ_A es llamada función de pertenencia y $\mu_A(x)$ es el grado de pertenencia de *x* a A. Cuanto más cerca de 1 es el valor de $\mu_A(x)$, *x* pertenece en mayor medida al conjunto A [18].

Un concepto clave en lo que sigue es el de alfa-corte, o corte de nivel alfa. Dado un conjunto difuso A y su función de pertenencia μ_A, el α-corte o corte de nivel α, simbolizado A_α, es el conjunto clásico definido como sigue:

$$A_\alpha = \{x \in X: \mu_A(x) \geq \alpha\} \qquad (2)$$

El valor α representa el nivel de *credibilidad* o de validez de que cada uno de los elementos *x* pertenezcan al conjunto A .

Una forma de obtener funciones de pertenencia difusa es mediante transformaciones probabilidad-posibilidad. Es bien sabido que los conjuntos difusos normales codifican distribuciones de posibilidad [18], de modo que es posible utilizar transformaciones entre otro tipo de medidas, por caso, las de probabilidad. La idea de estas transformaciones es conservar la incertidumbre modelada por las diferentes medidas, según el principio de invariancia de incertidumbre [19].

A. Relaciones difusas y similitud

Las relaciones difusas generalizan el concepto de relaciones de la misma forma que los conjuntos difusos generalizan la idea de conjuntos clásicos permitiendo una asociación parcial entre los elementos de un

universo. En este sentido, una relación binaria difusa R es caracterizada por una función de pertenencia μ_R: $X \times Y \rightarrow [0,1]$, interpretada como el grado de relación entre elementos pertenecientes a X y a Y. $\mu_R(x,y) = 1$ significa que los elementos están totalmente relacionados, y $\mu_R(x,y) = 0$ que no se relacionan en lo absoluto.

Si los conjuntos X e Y son finitos, entonces la relación difusa puede ser expresada como una matriz de orden $|X| \times |Y|$.

Cuando una relación difusa cumple con las propiedades de reflexividad y simetría se denomina relación de compatibilidad. Si además cumple con alguna clase de *-transitividad, entonces se denomina relación de equivalencia difusa o relación de similitud. Cabe destacar que es posible obtener una relación de equivalencia difusa a partir de una de compatibilidad mediante la clausura *-transitiva de esta [18].

Una particularidad de las relaciones de similitud es que todos sus alfa-cortes son relaciones de equivalencia clásicas y por lo tanto inducen partición dentro del conjunto en el que se definen. Cada una de estas particiones es un refinamiento de la anterior cuando se las ordena de forma descendente según alfa, de modo que las clases de equivalencia de un alfa-corte están incluidas en alguna otra clase de equivalencia del alfa-corte siguiente. En la figura 2 se puede ver el árbol de las particiones de la relación de equivalencia difusa definida en {a, b, c, d, e, f, g} dada por la matriz de la figura 1:

$$\widetilde{R} = \begin{bmatrix} 1 & 0.8 & 0 & 0.4 & 0 & 0 & 0 \\ 0.8 & 1 & 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0.9 & 0.5 \\ 0.4 & 0.4 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0.9 & 0.5 \\ 0 & 0 & 0.9 & 0 & 0.9 & 1 & 0.5 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0.5 & 1 \end{bmatrix}$$

Figura 1: Matriz de una relación de equivalencia difusa

En este árbol de particiones puede verse que los elementos que están más fuertemente relacionados son c y e (nivel α=1), luego se suma f a estos dos (nivel α=0.9), posteriormente a con b (nivel α=0.8), luego g se suma a c, e y f (nivel α=0.5), y, finalmente, d se suma a b y a (nivel α=0.4). Esta información también puede visualizarse especificando las clases de equivalencia nuevas que aparecen a medida que desciende el nivel alfa de la siguiente manera:

α=1

c, e

α=0.9

c, e, f

α=0.8

a, b

α=0.5

c, e, f, g

α=0.4

a, b, d

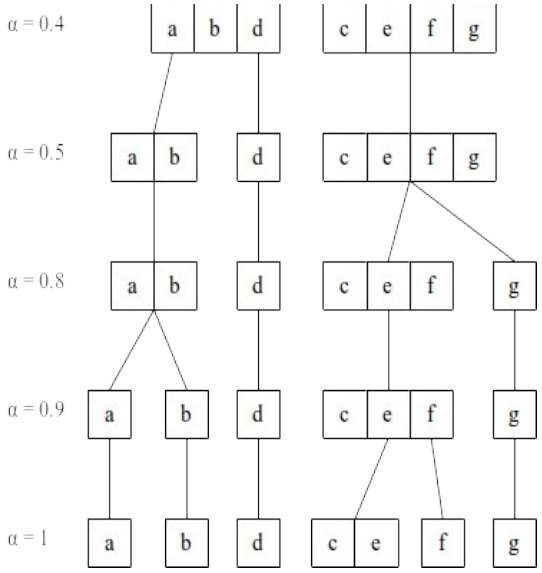

Figura 2: Árbol de particiones de una relación de equivalencia difusa

5. Análisis de Historias de Usuario mediante Lógica Difusa

A continuación se describe el modelo propuesto para el análisis de historias de usuario.

Se comienza construyendo un conjunto difuso por cada palabra. Este conjunto difuso se compone de pares ordenados de palabras que se encuentran a izquierda y derecha de la palabra en cuestión en cada texto. El grado de pertenencia resulta de una transformación probabilidad-posibilidad (la cual se explica más adelante). Formalmente, para cada palabra *w* en el texto, es posible construir un multiconjunto (o "tabla de frecuencias") a partir de todas las ocurrencias de *w* en los textos, consignando los pares de palabras que se encuentran a su inmediata izquierda y derecha, esto es, su **contexto**. Llamemos TF_w a tal multiconjunto, y llamemos $\mu_{TFw}(l, r)$ a la cantidad de ocurrencias de la

subcadena *l w r* en el corpus (esto es, el conjunto de historias de usuario). Luego, es posible calcular la probabilidad normalizando estas frecuencias y transformándolas en frecuencias relativas, de la siguiente manera:

$$Pr_w(l,r) = \frac{\mu_{TF_w}(l,r)}{|TF_w|} \quad (3)$$

donde $Pr_w(l, r)$ es la probabilidad de que l y r sean el contexto de w y $| TF_w |$ el cardinal del multiconjunto TF_w, es decir, la cantidad de ocurrencias de w en el corpus.

A partir de cada una de estas distribuciones de probabilidades es posible construir distribuciones de posibilidades mediante la fórmula [18]:

$$\pi_w(l,r) = \sum_{l',r' \in TF_w} min\left(Pr_w(l,r), Pr_w(l',r')\right) \quad (4)$$

Es así que cada palabra w en el texto posee un contexto difuso asociado

$$\widetilde{C}_w = \left\{ \frac{\pi_w(l,r)}{(l,r)} \right\} \quad (5)$$

A. Similitud entre palabras

A partir de los contextos difusos de cada palabra es posible obtener una medida de similitud entre palabras. Se propone una medida $WS(w_1, w_2)$ a partir de sus contextos difusos utilizando una extensión del índice de Jaccard para conjuntos clásicos [20], el cual se define como sigue:

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

De la misma manera, la similitud entre palabras $WS(w_1, w_2)$ toma en cuenta la similitud de los contextos de w_1 y w_2:

$$WS(w_1,w_2) = \frac{|\widetilde{C}_{w_1} \cap \widetilde{C}_{w_2}|}{|\widetilde{C}_{w_1} \cup \widetilde{C}_{w_2}|} \quad (7)$$

Puede verse claramente que esta medida resulta reflexiva ($WS(w,w) = 1$ para todo w) y simétrica ($WS(w_1,w_2) = WS(w_2,w_1)$ para cualesquiera w_1 y w_2), y por lo tanto, *WS* determina una relación difusa de compatibilidad entre palabras.

Es importante destacar que, ya que la cantidad de palabras es finita, puede construirse una matriz de similitud entre palabras a partir de la medida de similitud *WS*.

B. Similitud entre oraciones

Para comparar oraciones (o partes de oraciones) se utiliza la noción de similitud contextual entre palabras. Se supone que dos oraciones son más similares en tanto sus palabras son reemplazables unas por otras en su contexto. En particular, por cada palabra de la primera oración se busca cuál es la palabra más similar en la segunda oración, y se suman estos valores de similitud. Finalmente se divide la sumatoria por la cantidad de palabras en la primera oración.

$$SS_{\subseteq}(s_1,s_2) = \frac{\sum_{w_1 \in s_1} \operatorname{Sup}_{w_2 \in s_2} WS(w_1,w_2)}{|s_1|} \quad (8)$$

Esta medida se encuentra inspirada en la unificación semántica de FRIL [15].

La medida refleja la "inclusión" de las palabras de la primera oración en la segunda. Esta medida en general no es simétrica, pero se puede lograr la simetría mediante la fórmula:

$$SS_{=}(s_1,s_2) = SS_{\subseteq}(s_1,s_2) \wedge SS_{\subseteq}(s_2,s_1) \quad (9)$$

donde el operador de conjunción se traduce como el mínimo de ambos valores, de la manera usual.

De la misma manera que *WS*, $SS_=$ determina una relación de compatibilidad difusa, aunque esta última lo hace sobre el conjunto de oraciones.

C. Medición para cada criterio de calidad

En primera medida, dividimos cada HU en rol, medio y fin, y nos referimos a ellos como HU.rol, HU.medio y HU.fin respectivamente.

I. Ambigüedad

Como ya se dijo, la no ambigüedad consiste en evitar la utilización de palabras que tengan múltiples significados. En este sentido, cada fila de la matriz de similitud da el conjunto difuso de las palabras contextualmente similares a una palabra dada. Llamemos $SimWords_w$ a tal conjunto difuso. Es posible medir la incertidumbre presente mediante la incertidumbre-U:

$$U(SimWords_w) = \sum_{i=2}^{n} r_i \log_2 \frac{i}{i-1} \quad (10)$$

donde r_i surge de ordenar de manera descendente los valores de pertenencia del conjunto $SimWords_w$, de modo que $r_i \geq r_{i+1}$ para toda i. Esta incertidumbre-U es una medida de inespecificidad, la cual se relaciona con

conjuntos de alternativas, en este caso, la cantidad de palabras similares o posibles reemplazos que tiene la palabra *w* en todos sus contextos.

II. Conflicto y unicidad

Como ya se dijo más arriba, es posible comparar los contextos de cada palabra entre sí y calcular un índice de similitud de la oración. Particularmente si se divide la HU como se dijo se pueden calcular las distintas relaciones de unicidad/conflicto:

Duplicación total:

$SS_{=}$(HU1.rol, HU2.rol) ∧ $SS_{=}$(HU1.medio, HU2.medio) ∧ $SS_{=}$(HU1.fin, HU2.fin)

Diferentes medios, mismo fin:

1 - $SS_{=}$(HU1.medio, HU2.medio) ∧ $SS_{=}$(HU1.fin, HU2.fin)

Mismo medio, diferente fin:

$SS_{=}$(HU1.medio, HU2.medio) ∧ 1 - $SS_{=}$(HU1.fin, HU2.fin)

Cada una de estas expresiones han sido inspiradas por el *framework* QUS original, extendidas para tomar en consideración la similitud contextual.

6. Experimentación

A los fines de realizar una primera validación técnica del modelo propuesto se realizaron varios experimentos, los cuales se describen en esta sección. Todos ellos se realizaron a través de código escrito por los autores utilizando el lenguaje Python.

Es importante destacar en primer lugar que el análisis de textos posee múltiples herramientas, pero existen algunos pasos comunes que se encuentran en la gran mayoría de las aplicaciones de esta naturaleza. Lo primero es la conformación del conjunto de datos (*dataset*). En el análisis de textos la entidad de mayor jerarquía es el documento, que está compuesto por sentencias y a la vez estas están constituidas por palabras. En algunas aplicaciones el *dataset* está compuesto directamente de sentencias (como será en nuestro caso).

Como parte del aporte de este trabajo los autores recopilaron historias de usuario en español provenientes de diversas fuentes y se conformaron *datasets* que se encuentran disponibles en la web. Actualmente son muy escasos los *datasets* de historias de usuario, aún en idioma inglés. En la Tabla 1 se pueden ver los conjuntos de historias de usuario recopilados junto con información relativa a su tamaño y la cantidad de ellas que cuentan con HU.fin no vacío.

Tabla 1. Datasets recopilados

Nombre	Cantidad de HUs	Cantidad de HUs con Fin
HU 8	14	12
HU 9	26	3
HU 20	22	22
HU 90	35	3

Los conjuntos de datos no se reproducen aquí por razones de espacio, pero pueden encontrarse en [21].

A. Pre-procesamiento de los datasets

Al conformar el conjunto de datos pueden existir documentos en blanco (vacíos), términos con errores de ortografía, diferencias entre mayúsculas y minúsculas, términos de "relleno" (como conectores o pronombres) y signos de puntuación. Por esta razón se suelen "estandarizar" las palabras mediante algunas reglas generales que aplican a la gran mayoría de las situaciones: quitar las palabras de relleno, convertir todo lo restante a minúsculas, quitar todos los caracteres especiales (números, signos de puntuación) y realizar el *stemming* de las palabras restantes. Para comprender el proceso de *stemming*, se puede pensar en cada palabra como un árbol, con diferentes ramificaciones (por ejemplo, producida por prefijos, conjugaciones, cambios vocales). El *stemming* corta las ramas y se queda con el núcleo o raíz de la palabra. Por ejemplo, la palabra raíz "corre" reemplaza las ocurrencias de "corriendo", "correr", "corría", etc.

Puede que la sola cuenta de ocurrencias de palabras no sea adecuada como representativa de la importancia de una palabra en un *dataset*. Los conectores, artículos y otras expresiones (*stopwords*) pueden aparecer recurrentemente y sin embargo no resultan relevantes en la realización de un análisis de texto. Normalmente se las quita del corpus por esta razón.

B. Identificación manual de posibles defectos

A partir de las descripciones de los criterios de calidad se identificaron posibles defectos que deberían ser identificados por el modelo. Esta inspección se realizó manualmente antes de realizar el análisis mediante el modelo. En la Tabla 2 se detallan los defectos encontrados.

Tabla 2. Defectos encontrados

Grupo	Defectos
HU 8	Presencia de épicas puede dar problemas (¿distinto medio, mismo fin?) Igual fin
HU 9	La mayoría de las historias no tiene fin
HU 20	Posibles duplicados o ambigüedades - 20.1 y 20.2 - 20.7 y 20.8 - 20.21 (¿qué significa mejor/peor en este contexto?)
HU 90	La mayoría de las historias no tiene fin Posibles duplicados o ambigüedades - 90.4, 90.5, 90.6, 90.7, 90.8 - 90.33, 90.34 y 90.35

C. Análisis preliminar utilizando el modelo

Cada uno de los *datasets* fueron analizados inicialmente con el modelo propuesto, y se cotejaron los posibles defectos encontrados en la etapa anterior con los encontrados por la herramienta y sus niveles alfa. En la Tabla 3 se muestran la ambigüedad media y máxima de cada historia, junto a las 5 palabras (stems) más ambiguas. Además, en la Tabla 4 se muestran las posibles duplicaciones totales, con sus respectivos niveles de alfa, y mismo medio y distinto fin, y distinto medio y mismo fin (sólo los que superan el umbral de 0.5).

Tabla 3. Ambigüedades encontradas

ID	U Media	U Máxima	Palabras (stems) más ambiguas
HU8.1	0,16	1,58	gener
HU8.1.1	0,16	1,58	imprim
HU8.1.2	0,16	1,58	vend
HU8.2	0,04	0,13	predi
HU8.3	0,02	0,13	festival
HU8.4	0,04	0,13	
HU8.5	0,03	0,13	
HU8.5.1	0,02	0,13	
HU8.5.2	0,02	0,13	
HU8.6	0,04	0,13	
HU8.7	0,02	0,13	
HU8.8	0,01	0,13	
HU8.9	0,02	0,13	
HU8.10	0,01	0,13	
HU9.1	0,1	0,19	Modific
HU9.2	0,11	0,82	elimin
HU9.3	0,19	0,82	escog
HU9.4	0,02	0,19	observ
HU9.5	0,18	0,82	cantid
HU9.6	0,15	0,82	categor
HU9.7	0,07	0,15	
HU9.8	0,19	0,82	
HU9.9	0,25	0,82	
HU9.10	0,07	0,25	
HU9.11	0,24	0,82	
HU9.12	0,17	0,82	
HU9.13	0,02	0,15	
HU9.14	0,03	0,82	
HU9.15	0,02	0,15	
HU9.16	0,04	0,6	
HU9.17	0,09	0,25	
HU9.18	0,15	0,82	
HU9.19	0,15	0,82	
HU9.20	0,05	0,19	
HU9.21	0,02	0,19	
HU9.22	0,05	0,19	
HU9.23	0,11	0,6	
HU9.24	0,08	0,61	
HU9.25	0,03	0,19	
HU9.26	0,07	0,82	
HU20.1	0,19	0,7	sex
HU20.2	0,17	0,7	estructur
HU20.3	0,09	0,7	gener
HU20.4	0,12	0,7	ausenci
HU20.5	0,24	0,77	rrhh
HU20.6	0,27	0,77	administr
HU20.7	0,28	0,94	
HU20.8	0,29	1,01	
HU20.9	0,09	0,7	

HU20.10	0,08	0,7	
HU20.11	0,14	0,71	
HU20.12	0,16	0,71	
HU20.13	0,19	1	
HU20.14	0,12	0,71	
HU20.15	0,07	0,3	
HU20.16	0,1	1	
HU20.17	0,07	0,3	
HU20.18	0,07	0,33	
HU20.19	0,06	0,33	
HU20.20	0,03	0,3	
HU20.21	0,08	0,94	
HU20.22	0,03	0,7	
HU90.1	0	0	puent
HU90.2	0,09	0,45	bifurc
HU90.3	0,05	0,27	anill
HU90.4	0,21	1,58	1
HU90.5	0,21	1,58	3
HU90.6	0,21	1,58	fuert
HU90.7	0,22	1	
HU90.8	0,22	1	
HU90.9	0,08	0,27	
HU90.10	0,08	0,27	
HU90.11	0,11	0,45	
HU90.12	0,03	0,21	
HU90.13	0,02	0,21	
HU90.14	0,04	0,21	
HU90.15	0,03	0,21	
HU90.16	0,04	0,21	
HU90.17	0	0	
HU90.18	0,13	1	
HU90.19	0,03	0,21	
HU90.20	0,04	0,21	
HU90.21	0,04	0,21	
HU90.22	0,11	0,45	
HU90.23	0,04	0,21	
HU90.24	0,03	0,21	
HU90.25	0,03	0,21	
HU90.26	0,1	1	
HU90.27	0,02	0,21	
HU90.28	0,02	0,21	
HU90.29	0,1	0,79	
HU90.30	0,02	0,21	
HU90.31	0,11	0,45	
HU90.32	0,04	0,21	
HU90.33	0,2	1,29	
HU90.34	0,14	0,79	
HU90.35	0,2	1,29	

Tabla 4. Conflicto y unicidad

Grupo	Defectos
HU 8	*Duplicados* α=1 HU8.1, HU8.1.1, HU8.1.2 α=0.67 HU8.5.1, HU8.5.2 α=0.5 HU8.5, HU8.5.1, HU8.5.2 α=0.25 HU8.4, HU8.5, HU8.5.1, HU8.5.2 α=0.167 HU8.4, HU8.5, HU8.5.1, HU8.5.2, HU8.8 *Distinto medio, igual fin* HU8.6 y HU8.7 (1) *Igual medio, distinto fin* HU8.2 y HU8.3 (1) HU8.9 y HU8.10 (0.6)
HU 9	*Duplicados* α=0.95 HU9.18 y HU9.19 α=0.89 HU9.17, HU9.18 y HU9.19 α=0.63 HU9.8 y HU9.9 α=0.57 HU9.11 y HU9.12 α=0.525 HU9.1 y HU9.3 α=0.524 HU9.5, HU9.8 y HU9.8
HU 20	*Duplicados* α=0.88 HU20.7, HU20.8

	α=0.66 HU20.5, HU20.6 α=0.63 HU20.5, HU20.6, HU20.7, HU20.8 *Distinto medio, igual fin* HU20.1 y HU20.2 (0.67) HU20.2 y HU20.3 (0.6) HU20.3 y HU20.12 (0.6) *Igual medio, distinto fin* HU20.11 y HU20.13 (0.7) HU20.5 y HU20.11 (0.67) HU20.5 y HU20.13 (0.67) HU20.7 y HU20.11 (0.67) HU20.7 y HU20.13 (0.67) HU20.8 y HU20.11 (0.67) HU20.8 y HU20.13 (0.67) HU20.11 y HU20.12 (0.67)
HU 90	*Duplicados* α=1 HU90.4, HU90.5, HU90.6 HU90.7, HU90.8 HU90.33, HU90.35 α=0.93 HU90.33, HU90.34, HU90.35 α=0.66 HU90.16, HU90.17 α=0.5 HU90.19, HU90.20

Para HU 8, los niveles de duplicación total de menos de 0.5 no constituyen duplicados. Además, el falso positivo de *Distinto medio, igual fin* se debe a la ausencia de fin (la medida de similitud asigna 1 por defecto ante cadenas vacías). Finalmente, 8.2 y 8.3 efectivamente poseen el mismo medio y distinto fin por ser parte de una épica, mientras que si bien 8.9 y 8.10 están relacionados, no constituyen un defecto.

Para HU 9 se omitieron los demás indicadores y sólo se consignó la duplicación total. Todo lo detectado es falso positivo, aunque las historias de usuario que asocia están muy relacionadas.

Para HU 20 se detectaron las historias 20.7 y 20.8 como las más ambigüas, 20.1 y 20.2 aparecen con niveles altos, y 20.5, 20.6 y 20.13 aparecen dentro de las más ambiguas, aunque no habían sido etiquetadas manualmente. Además, si bien 20.21 no aparece dentro de las más ambiguas, sí cuenta con al menos una palabra muy ambigua (0.94). Por el lado de los duplicados, 20.7 y 20.8 son detectados primero, mientras que 20.5 y 20.6 aparecen después, aunque no habían sido etiquetados como duplicados. 20.1 y 20.2 no aparecen como duplicados, aunque sí en Distinto medio, igual fin como los primeros. Los demás no habían sido etiquetados originalmente y no constituyen defectos de calidad.

Finalmente, el grupo HU 90 posee características similares a HU 9, por lo que sólo se consignan los resultados de duplicación total en la parte de conflicto y unicidad. Las historias más ambiguas son las comprendidas entre la HU90.4 y la HU90.8 inclusive, seguidas de HU90.33, HU90.35 y HU90.34, estas últimas sí habían sido etiquetadas manualmente como ambiguas. También son registradas por el modelo como muy similares entre sí. Luego aparecen algunas otras más con menor similitud, no etiquetadas originalmente, aunque relacionadas funcionalmente.

Puede apreciarse, en consecuencia, que los falsos positivos encontrados por duplicación total corresponden mayormente a historias de usuario muy relacionadas entre sí, ya sea por cubrir funcionalidad similar como por formar parte de una épica, o por palabras que resultan contextualmente ambiguas.

D. Eliminación de defectos

Los posibles defectos encontrados por el modelo fueron abordados por un equipo para intentar eliminarlos, teniendo en cuenta las definiciones de los criterios de calidad de QUS. Se trabajó sobre el conjunto HU 9 con la idea de agregar algunos fines faltantes, de modo de poder establecer mayores diferencias entre las HUs registradas como similares por el modelo. Posteriormente se volvió a analizar el conjunto de historias utilizando el modelo para evaluar el comportamiento del mismo.

En la nueva ejecución del modelo todos los valores de duplicación total cayeron por debajo de 0.46, y las primera que aparecen son HU9.9 y HU9.11, que se dejaron sin un fin. Esto abona la hipótesis de que los fines de las historias de usuario ayudan a evitar los falsos positivos. Por otra parte, los indicadores “Mismo medio, distinto fin” y “Distinto medio, mismo fin” no arrojan valores de similitud superiores a 0.5 para aquellas historias modificadas.

El trabajo realizado en esta sección constituye una primera demostración de abordaje de resolución de defectos de calidad: particularmente los falsos positivos correspondientes a historias que no tienen especificado un fin pueden eliminarse escribiendo un fin adecuado para tales historias, el cual refleje las distintas necesidades e intenciones del usuario al momento de utilizar el software.

E. Inserción de defectos

A los fines de profundizar el análisis los autores introdujeron defectos de calidad en el grupo HU 8

siguiendo las definiciones de QUS. Estos defectos conocidos debían ser detectados por el modelo como fallas de calidad en alguno de los criterios considerados. Concretamente se agregaron 3 historias de usuario que resultan ser muy parecidas a HU8.3, HU8.6 y HU8.7, pero renarradas.

HU8.6.COPIA: *Como Responsable de Festival quiero poder establecer precios de los festivales*.

HU8.3.COPIA *Como Responsable de Festival quiero planificar un festival para elegir los grupos musicales que actuarán cada día*.

HU8.7.COPIA *Como Responsable de Predio quiero elegir las butacas para las que luego podré vender entradas*.

Los resultados del modelo arrojan que en duplicación total se asocia a HU8.3 con su copia a nivel 0.5, y a los restantes con su copia a nivel 0.25. Los demás indicadores tampoco indican nada de relevancia. Este último resultado debe mejorarse a fin de poder detectar similitudes más semánticas con el modelo.

7. Conclusiones y trabajos futuros

En este trabajo se ha presentado la importancia de la calidad de los entregables en el proceso de ingeniería de software, así como la dificultad para determinarla y los inconvenientes posteriores que produce en el proceso, pudiendo ser un causal de fracaso de un proyecto.

En tal sentido, se ha propuesto un modelo de análisis de historias de usuario siguiendo la línea y mejorando el framework QUS (*Quality User Story*), introduciendo medidas de similitud contextual que no se hallan presentes en la formulación original. El formalismo utilizado para tal fin es el de la Lógica Difusa, de amplia trayectoria en el manejo de ambigüedad y otros tipos de incertidumbre en el lenguaje natural. El resultado es un modelo de análisis de texto que determina una medición de la calidad de un conjunto de historias de usuario en tres características clave: no ambigüedad, ausencia de conflictos y unicidad.

El comportamiento del modelo fue demostrado inicialmente en cuatro conjuntos de historias de usuario recopiladas por los autores provenientes de diversas fuentes. Se realizaron distintos experimentos para detectar las fortalezas y debilidades del modelo, y pudo observarse que el modelo es capaz de detectar la mayor parte de los defectos de calidad presentes en cada conjunto de historias, aunque también es propenso a los falsos positivos, y pierde eficacia al intentar detectar duplicaciones en historias demasiado diferentes, aunque semánticamente equivalentes.

Pudo observarse también que las HU muy relacionadas por ser parte de una épica pueden resultar en falsos positivos en duplicación total, sobre todo en aquellas que no poseen especificado su fin. Este último hecho se puede paliar agregándolo, lo cual se pudo verificar mediante experimentación.

El modelo, por lo tanto, resulta en un valioso aporte, especialmente en contextos como el de las PyMEs, que suelen caracterizarse por la escasez de recursos y la superposición de roles.

Como trabajos futuros se espera poder realizar una validación más exhaustiva comparando el modelo propuesto con otro tipo de técnicas de análisis de texto, o hasta combinando enfoques más semánticos, basados en corpus o en ontologías, para poder superar las dificultades que pudieron detectarse.

Agradecimientos

Gracias a la Universidad Autónoma de Entre Ríos a través del PIDAC *Técnicas para la mejora de la calidad en la Ingeniería de Requisitos en las empresas de software de Argentina*, y a la Universidad Tecnológica Nacional a través del proyecto SIUTICU0005297TC *Enfoques de Optimización Multiobjetivo basados en Preferencias en la Ingeniería de Software*. Asimismo, gracias a las colaboradores que participaron en la recolección de Historias de Usuario del dataset: Viviana Bourdetta y Laura Helena Ríos.

Referencias

[1] F. Barletta, M. Pereira, y G. Yoguel, «Impacto de la Política de Apoyo a la Industria de Software y Servicios Informáticos», 2014. Accedido: mar. 05, 2019. [En línea]. Disponible en: http://www.ciecti.org.ar/wp-content/uploads/2016/05/DT4-SSI_v3.pdf.

[2] S. de la T. P. Dirección Nacional de Análisis y Estadísticas Productivas, Subsecretaría de Desarrollo y Planeamiento Productivo, «Argentina productiva: economía del conocimiento», 2019. [En línea]. Disponible en: https://biblioteca.produccion.gob.ar/document/download/500.

[3] A. Hernández Paez, Y. Fuentes Castillo, y J. A. Santos, «Buenas Prácticas para el Desarrollo de Requisitos

basado en Componentes utilizando el Modelo de Capacidad y Madurez Integrada», *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 14, n.º 1, pp. 215-225, 2021.

[4] T. Clancy, «The standish group chaos report», 2014.

[5] E. Kheirkhah y A. Deraman, «Important factors in selecting requirements engineering techniques», en *Proceedings - International Symposium on Information Technology 2008, ITSim*, 2008, vol. 3, doi: 10.1109/ITSIM.2008.4631895.

[6] J. Medeiros, A. Vasconcelos, C. Silva, y M. Goulão, «Quality of software requirements specification in agile projects: A cross-case analysis of six companies», *Journal of Systems and Software*, vol. 142, pp. 171-194, ago. 2018, doi: 10.1016/J.JSS.2018.04.064.

[7] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, y S. Brinkkemper, «Improving agile requirements: the Quality User Story framework and tool», *Requirements Engineering*, vol. 21, n.º 3, pp. 383-403, sep. 2016, doi: 10.1007/S00766-016-0250-X/TABLES/5.

[8] B. Wake, «INVEST in good stories, and SMART tasks», *XP123*. 2003.

[9] I. C. S. S. E. S. Committee y I.-S. S. Board, *IEEE recommended practice for software requirements specifications*, vol. 830, n.º 1998. IEEE, 1998.

[10] M. Glinz, «Improving the Quality of Requirements with Refactoring», en *Proceedings of the Second World Congress for Software Quality (2WCSQ)*, 2000, pp. 55-60.

[11] P. Heck y A. Zaidman, «A Quality Framework for Agile Requirements: A Practitioner's Perspective», jun. 2014, Accedido: sep. 13, 2022. [En línea]. Disponible en: http://arxiv.org/abs/1406.4692.

[12] D. Berry, R. Gacitua, P. Sawyer, S. T.-I. working conference, y undefined 2012, «The case for dumb requirements engineering tools», *Springer*, 2012, Accedido: sep. 13, 2022. [En línea]. Disponible en: https://link.springer.com/chapter/10.1007/978-3-642-28714-5_18.

[13] D. M. Berry y E. Kamsties, «Ambiguity in Requirements Specification», *Perspectives on Software Requirements*, pp. 7-44, 2004, doi: 10.1007/978-1-4615-0465-8_2.

[14] T. Kenter y M. De Rijke, «Short text similarity with word embeddings», *International Conference on Information and Knowledge Management, Proceedings*, vol. 19-23-Oct-2015, pp. 1411-1420, oct. 2015, doi: 10.1145/2806416.2806475.

[15] M. Azmi-Murad y T. P. Martin, «Using fuzzy sets in contextual word similarity», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3177, pp. 517-522, 2004, doi: 10.1007/978-3-540-28651-6_76.

[16] P. Bafna, D. Pramod, y A. Vaidya, «Document clustering: TF-IDF approach», *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, pp. 61-66, nov. 2016, doi: 10.1109/ICEEOT.2016.7754750.

[17] L. A. Zadeh, «Fuzzy Logic = Computing with Words», pp. 3-23, 1999, doi: 10.1007/978-3-7908-1873-4_1.

[18] G. J. Klir y B. Yuan, *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall, 1995.

[19] G. J. Klir, *Uncertainty and Information: Foundations of Generalized Information Theory*. 2005.

[20] V. Zhelezniak, A. Savkov, A. Shen, F. Moramarco, J. Flann, y N. Y. Hammerla, «Don't Settle for Average, Go for the Max: Fuzzy Sets and Max-Pooled Word Vectors», 2019, [En línea]. Disponible en: https://openreview.net/forum?id=SkxXg2C5FX.

[21] C. Casanova, K. Cedaro, y R. Sosa Zitto, «Spanish User Story Dataset», *Mendeley Data*. 2022, doi: 10.17632/ws6hjmjmh7.3.