//UCIENCIA/23
CONVENCIÓN CIENTÍFICA INTERNACIONAL

Ambigüedad léxica en requisitos de software: Un mapeo sistemático

Lexical ambiguity in software requirements: A systematic mapping

[1*] Samira Enriquez González , Viana Gabriela Jacomino Díaz [2], Dunia María Colomé Cedeño [3]

[1] Universidad de las Ciencias Informáticas. samiradlmeg@uci.cu
[2] Universidad de las Ciencias Informáticas. vianagjd@estudiantes.uci.cu
[3] Universidad de las Ciencias Informáticas. dcolome@uci.cu

[*] Autor para correspondencia: samiradlmeg@uci.cu

Resumen

La ingeniería de requisitos es una de las etapas más importantes del ciclo de vida del desarrollo de software. El éxito de cualquier producto de software depende de la calidad de sus requisitos. Los requisitos de software suelen estar escritos en lenguaje natural. La ambigüedad en los requisitos escritos en lenguaje natural es un problema que ha sido estudiado por la comunidad de ingeniería de requisitos durante más de dos décadas. La resolución manual de la ambigüedad en los requisitos es trabajosa y requiere de mucho tiempo. Este artículo examina el panorama de la investigación sobre la ambigüedad léxica presente en los requisitos de software, para comprender su estado e identificar los problemas pendientes. Para llevar a cabo este trabajo se ha utilizado el método de estudio de mapeo sistemático. Se han identificado 30 estudios de interés según el contexto de la investigación y se han revisado con el uso de tres preguntas de investigación, que abarcan cinco aspectos de la investigación sobre la ambigüedad léxica en los requisitos de software: el estado de la bibliografía, el estado de la investigación empírica, el enfoque de la investigación, el estado de la práctica y las tecnologías utilizadas. Como resultado de este estudio se puede decir que las investigaciones analizadas resuelven la tarea de detección, el 92% de los estudios tratan el requisito en bruto mientras que el 8% trata el requisito en bruto y el documento de especificación de requisitos. De los estudios seleccionados se han propuesto 21 nuevas herramientas para apoyar tareas de análisis lingüístico, seis técnicas, las tecnologías más utilizadas son Procesamiento del Lenguaje Natural, ontología, Lenguaje Natural Controlado y web semántica.

Palabras clave: ambigüedad léxica, mapeo sistemático, requisitos.

Abstract

Requirements engineering is one of the most important stages of the software development life cycle. The success of any software product depends on the quality of its requirements. Software requirements are usually written in natural language. Ambiguity in requirements written in natural language is a problem that has been studied by the requirements engineering community for more than two decades. Manually resolving ambiguity in requirements is laborious and time consuming. This article examines the research landscape on lexical ambiguity present in software requirements, to understand its status and to identify outstanding issues. To carry out this work, the systematic mapping study method has been used. 30 studies of interest have been identified based on the research context and have been reviewed using three research questions, covering five aspects of research on lexical ambiguity in software requirements: the state of the literature, the state of empirical research, research focus, state of practice, and technologies used. As a result of this study, it can be said that the analyzed investigations solve the detection task, 92% of the studies deal with the raw requirement while 8% deal with the raw requirement and the requirements specification document. Of the selected studies, 21 new tools have been proposed to support linguistic analysis tasks, six techniques, the most used technologies are Natural Language Processing, ontology, Controlled Natural Language and semantic web.

Keywords: *lexical ambiguity, requirements, systematic mapping.*

I. Introducción

La ambigüedad es una de las principales causas de los fracasos de la etapa de Ingeniería de Requisitos (IR). Llevar a cabo de manera adecuada el proceso de IR disminuye la probabilidad de fracaso de un proyecto. Los requisitos bien definidos permiten conocer de un modo conciso lo que debe ser capaz de realizar el software a desarrollar además de orientar las actividades, recursos y esfuerzos de manera eficiente permitiendo la disminución de costos y retrasos. Dentro de la IR se realizan las tareas educción, documentación, validación y gestión de requisitos (Aschauer, 2018). Documentar los requisitos tal y como lo describe el cliente, de modo que todas las partes interesadas comprendan el contexto exacto de los requisitos forma parte de la tarea fundamental de la IR (Sabriye et al., 2018). La especificación de requisitos es una de las actividades que se realiza dentro de la documentación. Se debe elaborar una especificación

de calidad porque constituye la base del desarrollo de software y la misma trae consigo un software de calidad. Una mala especificación conlleva a una pérdida de tiempo o al fracaso del producto, esto ocurre porque la especificación de requisito generalmente se realiza en lenguaje natural (LN). Debido a ello se hace probable la aparición de varios defectos como la ambigüedad en los requisitos.

Un requisito ambiguo es un requisito que tiene diferentes significados y se puede interpretar de múltiples formas dependiendo de la ambigüedad que posea. La ambigüedad es como una anomalía, pero también, como un fenómeno natural que penetra todo el lenguaje. Hay que relacionarla y a la vez diferenciarla de la vaguedad y del lenguaje metafórico. En todas estas manifestaciones existen multiplicidad de significados y a la vez la exigencia de seleccionar uno y prescindir de los restantes. La ambigüedad supone la existencia de dos o más significados excluyentes entre sí, pero bien definidos, mientras que la vaguedad se basa en la indeterminación del referente y el lenguaje metafórico establece una relación entre el plano real y el metafórico (Osama et al., 2020).

Dentro de los tipos de ambigüedad está la lingüística, que se puede definir como la posibilidad de encontrar dos o más significados para una misma serie de estímulos grafémicas, palabras, enunciado o texto (Osama et al., 2020). Algunas de las ambigüedades lingüística que presentan los requisitos de software son ambigüedad pragmática, semántica, sintáctica ponderadas, polisemia y léxica que es la ambigüedad a tratar en este estudio (Osama et al., 2020).

La ambigüedad léxica se refiere a las palabras o frases que tienden a tener más de un significado en dependencia del contexto, como es el caso de cubo-Figura geométrico y cubo-recipiente. Este tipo de ambigüedad puede clasificarse en homónima y polisémica. La homonimia se produce cuando varias palabras tienen la misma ortografía o fonética, pero significados diferentes. Por ejemplo, lo que sucede con llama-fuego y llama-animal, que derivan del latín. Por otro lado, la polisemia se produce cuando una misma palabra tiene diferentes significados en distintos contextos. Esto se evidencia con la palabra periódico que puede significar 1) publicación física impresa “el periódico se mojó con la lluvia”, 2) la ilustración editora “el periódico filmó a su personal de edición” (Osama et al., 2020) (Zhao, 2022). Para la detección y reducción de la ambigüedad en la documentación de requisitos, Hayman (2018) identificó distintas soluciones que pueden mejorar la calidad de la especificación de requisitos. Dentro de ellas se encuentran las placas de caldera, inspecciones, ontologías, los lenguajes controlados y el procesamiento del lenguaje natural (PLN).

Se han realizado numerosos estudios sobre la ambigüedad de los requisitos en lenguaje natural. En este artículo, se presentan los resultados de un estudio de mapeo sistemático, en un intento de descubrir los trabajos empíricos de las dos últimas décadas que abordan la detección de ambigüedad léxica en los requisitos de software con la ayuda de

herramientas y técnicas, así como métodos y algoritmos. Este estudio de mapeo se ha realizado utilizando las directrices de la revisión sistemática de la literatura, de la Ingeniería del Software Basada en la Evidencia. Los estudios de mapeo sistemático son estudios secundarios recomendados para proporcionar un resumen visual o un mapa de grano grueso de la investigación empírica sobre un tema (Zhao, 2021). El objetivo de este tipo de estudio es identificar lagunas en un conjunto de estudios primarios y los resultados obtenidos permiten comprender el estado del tema en ámbito de investigación y son útiles para diseñar el marco analítico de las revisiones sistemáticas completas.

II. Materiales y método de revisión

2. Método de revisión

El mapeo sistemático se ha realizado utilizando el método presentado por Petersen, Vakkalanka y Kuzniarz (2015). Esta sección presenta las preguntas de investigación y describe las principales actividades involucradas en el mapeo; los resultados del mapeo se informan en la Sección III.

2.1. Preguntas de investigación

Las preguntas de investigación (PI) para este estudio de mapeo se indican a continuación. Las PI están interrelacionadas, diseñadas para analizar la literatura sobre ambigüedad léxica progresivamente. En primer lugar, describimos las principales PI en cursiva, y luego informamos en texto regular las preguntas específicas que se utilizan para elaborar cada PI. Las respuestas al conjunto de preguntas específicas proporcionan la respuesta a la PI principal.

P1: *¿Cuál es el estado de la literatura sobre ambigüedad léxica?* ¿Específicamente, cuál es la población de la literatura publicada sobre ambigüedad léxica? ¿Cuál es la línea de tiempo de las publicaciones? ¿Cuáles son los principales lugares de publicación?

P2: *¿Cuál es el estado de desarrollo de herramientas para la detección de ambigüedad?* ¿Qué nuevas herramientas han sido desarrolladas? ¿Cuál de estas herramientas están disponibles para su uso?

P3: *¿Qué tipo de tecnologías se han utilizado para el tratamiento de la ambigüedad en requisitos de software?*

2.2. Proceso de selección de estudios

2.2.1. Determinación de las fuentes de datos

Se identificaron las siguientes bases de datos digitales como las principales fuentes de datos para el estudio de mapeo: ACM Digital Library (ACM), Biblioteca digital IEEE Xplore, Scopus, Google académico y Springer. Se eligieron estas bibliotecas porque albergan las principales revistas y actas de conferencias relacionadas con la ingeniería de software y la ingeniería de requisitos.

2.2.2. Formulación de la estrategia de búsqueda

La estrategia de búsqueda se basó en la búsqueda directa de las bases de datos electrónicas de las bibliotecas digitales antes mencionadas. Los términos de búsqueda para consultar estas bibliotecas se construyeron siguiendo los pasos presentados en (Kitchenham et al., 2015). Específicamente, se emplearon los términos principales "ingeniería de requisitos" (que representa el contexto de la investigación), "tecnologías" (que representa la intervención de cualquier solución en este contexto) y "ambigüedad" (que es el problema a tratar en el contexto de la investigación) como términos base; se elaboró cada término base con grafías alternativas y sinónimos; se empleó el conector booleano OR para incorporar sinónimos, ortografías alternativas, términos alternativos y términos de subcampo en cada conjunto de términos base, y el AND para vincular los dos conjuntos de términos. Se realizaron varias iteraciones para identificar y refinar las palabras clave. El conjunto completo de los términos de búsqueda se presenta en la Tabla 1.

Tabla 1: Palabras clave

Palabras clave	**Palabras clave derivadas**
Ambiguity	Lexical ambiguity, disambiguation, ambiguity detection
Engineering requirements	requirements, requirements specification document, requirements elicitation
Technologies	tools, algorithms, methods, techniques

2.2.3. Búsqueda de literatura

Luego de una serie de búsquedas iniciales para afinar los términos de búsqueda, se realizó la búsqueda principal en el rango de fecha del 2015 al 2023 en las bibliotecas digitales antes mencionadas y se utilizó el rango de fecha del 2019 al 2023 en google académico. Para todas las bibliotecas, se realizó una búsqueda avanzada y solo se recuperaron las publicaciones escritas en inglés. Los resultados de la búsqueda se importaron a Mendeley con el objetivo de facilitar la aplicación de los criterios de inclusión y exclusión. Los resultados de la búsqueda inicial fueron un total de 126 y después de eliminar automáticamente los duplicados, los resultados totales fueron 110 como muestra la Figura 1. Para guiar la selección de artículos de interés investigativo se definió un conjunto de criterios de inclusión y exclusión, estos criterios se muestran en la Tabla 2. Los criterios excluyen explícitamente los libros, tutoriales, revistas, además de los artículos cortos porque en general tales artículos carecen de una descripción detallada de sus contribuciones.

Tabla 2: Selección de los estudios relevantes

I/E	No.	Criterio
E	1	Los artículos donde el resumen, palabras claves, introducción y conclusiones no tengan que ver con el contexto de la investigación.
E	2	Los artículos que traten la ambigüedad desde el punto de vista de la lingüística y artículos de revisión sistemática.
E	3	Los artículos que tengan al menos 2 páginas y que no entren en el rango de fecha del 2015 -2023.
E	4	Los resultados que sean libros, revistas y sitios web.
E	5	Los artículos que no ofrecen solución para la ambigüedad léxica.
I	1	Incluir estudios primarios revisados por pares que sean relevantes para solucionar la ambigüedad en la ingeniería de requisitos (es necesario realizar comprobaciones cruzadas y validaciones de dichos estudios).

I	2	Si hay varios estudios relevantes que informan sobre la misma investigación, incluir sólo el estudio más largo y excluir el resto.

Basándose en estos criterios, se llevó a cabo la selección de los estudios. El proceso de selección de estudios, está representado en la Figura 1.

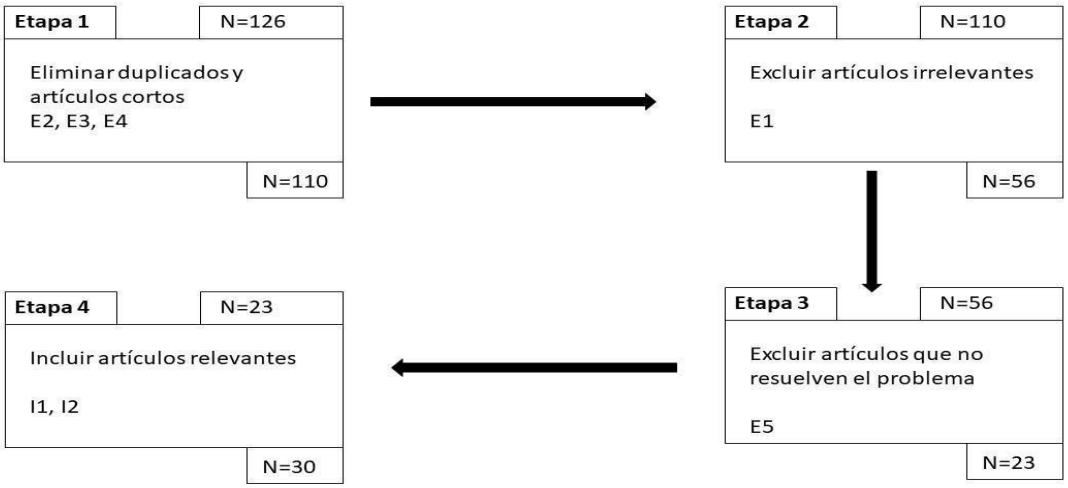

Figura 1: Selección de estudios.

Una vez obtenidos los resultados de búsqueda en las bases de datos mencionadas anteriormente, se aplicó una etapa de eliminación de duplicados. Luego de esta etapa se aplicaron los criterios de exclusión E2, E3 y E4 (Tabla 2) a los resultados de la búsqueda, para eliminar contenidos irrelevantes como libros, revistas y sitios web. También se realizó una serie de comprobaciones para asegurarse de que no quedaran artículos cortos en la biblioteca. Tras esta etapa se pasó de 110 a 56 artículos en el proceso. En el siguiente paso, se comprobó el título, el resumen, la introducción y conclusión de cada estudio, y se eliminaron los irrelevantes y los estudios secundarios de acuerdo con E1 y E5 donde de 56 se obtuvieron 23. En esta última etapa se revisaron de forma independiente los estudios de acuerdo con I1 y I2 para determinar su inclusión o no. Para ello fue necesario leer detenidamente el texto completo de cada estudio para establecer su pertinencia e identificar sus componentes clave con respecto a las categorías predefinidas. Al finalizar el

proceso de selección de estudios se obtuvo como resultado un total de 30 artículos de interés según el contexto de la investigación. La Tabla 3 muestra los resultados de la búsqueda y selección.

Tabla 3: Resultado de la búsqueda.

BASES DE DATOS	*TOTAL DE RESULTADOS*	*INTERESANTES*
SCOPUS	26	11
ACM	44	2
IEEE	11	10
Springer	3	3
Google académico	42	4
TOTAL	126	30

La Figura 2 ofrece una visión general de las publicaciones desde 2015 -2023 indicando el número de publicaciones sobre el tema.

Figura 2: Resultados de la búsqueda por año.

III. Resultados y discusión

Las Tablas 4, 5, 6 y 7 ofrecen una visión general de la variedad de técnicas, herramientas, métodos y recursos que han empleado los 30 estudios incluidos. Los 30 artículos de interés se centraron en más de un aspecto del tratamiento de la ambigüedad, por ejemplo, detección, reducción, eliminación y sugerir al ingeniero de requisitos. De los 30 estudios seleccionados, 25 se centraron en la detección de la ambigüedad, mientras que 5 de ellos también abordaron la reducción de la ambigüedad y solo uno detecta y brinda sugerencias de calidad. Los 30 artículos de interés abordan la ambigüedad léxica, pero 1/30 trata la ambigüedad heterónima y homográfica, 1/30 la sinónima y sobrecargada (homónima y polisémica a la vez), 30/30 homónima y polisémica.

Tabla 4: Técnicas encontradas

Técnicas	Referencia
Etiquetado POS y diccionario con términos ambiguos	(Aliisse & Hassan, 2019) (Zhao, 2022) (Benedikt et al., 2015)
Coincidencia de expresiones regulares y etiquetado POS	(Aggarwal & Rani, 2019)
WEB Semántica	(Zait & Zarour, 2019) (Maulud & Dastan, 2021)
Minería	(Husain, 2021)
Ontología	(Maulud & Dastan, 2021) (Bhatia et al., 2016)
Tokenización	(Zhao, 2022)

Tabla 5: Herramientas encontradas

Herramientas	Referencia
Gamify4Req (basada en reglas)	(Dar et al., 2022) (Hsdar et al., 2022)
Spacy	(Fantechi, 2021)
Quod	(Ferrari et al., 2019)
Qualice, RQA, Tiger Pro	(Lucassen & Arendse, 2016)
Quars	(Gnesi & Gianluca, 2016)
RQA, QVScribe, Quod, Innoslate, Rat	(Naeem et al., 2019) (Moreno et al., 2015)
Ambidexter, CKCO, SREE	(Yadav et al., 2021)
StanfordNLP, NLTK, OpenNLP, SPACY, Gate, DODT	(Chetan et al., 2015) (Schmitt X, 2022) (Stálhane & Wien, 2015)

Tabla 6: Métodos y algoritmos

Métodos y Algoritmos	Referencias
C-value	(Ywang, 2015)
N grama	(Husain, 2021)
Clustering,single link	(Ywang, 2015)
Clonalg	(Husain, 2021)

método **manual (inspección y revisión)**	(Raikar & Cholli N. G, 2021)
método semiautomático con PLN **(ontología y Patrones LN)**	(Raikar & Cholli N. G, 2021)
método **semiautomático** con aprendizaje automático **(árbol de decisión, support vector machine, naive bayes, N_ gram)**	(Raikar & Cholli N. G, 2021)
árbol de decisión, tacones y tablas de decisión, bosques aleatorios	(Osama et al., 2018)

Tabla 7: Recursos

Recursos	**Referencias**
WordNet	(Husain, 2021) (Raikar & Cholli N. G, 2021) (Bäumer et al., 2018) (Yadav et al., 2021)
ConcepNet, ResearchCye y Yaga	(Raikar & Cholli N. G, 2021)
Diccionarios	(Gnesi, 2019)
SBVR	(Yadav et al., 2021)

IV. Conclusiones

Los resultados de este estudio bibliográfico han arrojado las siguientes conclusiones:

- La comunidad de la ingeniería de requisitos no ha prestado suficiente atención a la evaluación empírica de las herramientas y técnicas para abordar la ambigüedad léxica en los requisitos de software.
- Los investigadores se han centrado más en la detección de la ambigüedad que en la resolución.
- Las ambigüedades homónima y polisémica son más abordadas para la evaluación empírica que otros tipos de ambigüedad léxica que pueden ser difíciles de detectar o resolver.

- Existen muy pocos resultados sobre la detección, resolución o eliminación de la ambigüedad léxica.
- Faltan investigaciones empíricas centradas específicamente en el análisis comparativo de las herramientas y técnicas para el tratamiento de la ambigüedad. Los 30 estudios de interés utilizan en su mayoría el PLN.
- Los estudios seleccionados han propuesto un total de 21 nuevas herramientas de apoyo a una serie de tareas de análisis lingüístico, pero hay pocas pruebas de que estas herramientas hayan sido adoptadas o aceptadas por la industria, lo que indica una falta de práctica industrial de los resultados de la investigación.

V. Referencias

1. Aggarwal, G., & Rani, A. (2019). *Algorithm for automatic detection of ambiguities from software requirements*. http://doi 10.35940/ijitee.I1141.0789S19
2. Aliisse, A., & Hassan, S. (2019). *A tool for detecting ambiguity in software requirements specification*. https://www.scopus.com/inward/record.uri?eid=2-s2.0
3. Aschauer, B. (2018). *IREB Certified Professional for Requirements Engineering-Advanced Level RE_Agile_Syllabus*.
4. Bäumer, Frederik, S., Geierhos, & Michaela. (2018). *Flexible ambiguity resolution and incompleteness detection in requirements descriptions via an indicator-based configuration of text analysis pipelines*.
5. Benedikt, G., Creighton, O., & Kof, L. (2015). *Ambiguity Detection: Towards a Tool Explaining Ambiguity Sources*.
6. Bhatia, M. P. S., Kumar, A., & Beniwal, R. (2016). *Ontology based framework for detecting ambiguities in software requirements specification*.
7. Chetan, A., Mehrdad, S., & Briand, L. (2015). *Automated Checking of Requirements Templates using Natural Language Processing*.

8. Dar, Hafsa, S., Imtiaz, S., Muhammad, L., & Ikram, U. (2022). *Design of gamification tools to reduce the ambiguity of the requirements during Elicitation*. http://doi10.1109/ICCI54321.2022.9756083
9. Fantechi, A. (2021). *A spaCy-based tool for extracting variability from NL requirements*. https://doi10.1016/j.aiopen.2021.05.001
10. Ferrari, A., Spagnolo, G., Fiscella, A., & Parente, G. (2019). *QuOD: An NLP Tool to Improve the Quality of Business Process Descriptions*. https://doi.org/10.1007/978-3-030309855_17
11. Gnesi, S., & Gianluca, T. (2016). *QuARS a NLP Tool for Requirements Analysis*.
12. Hayman, O. (2018). *An Analysis of Ambiguity Detection Techniques for Software Requirements Specification (SRS)*. http:// www.sciencepubco.com/index.php/IJET
13. Hsdar, S., Imtiaz, & Ullah, L. (2022). *Gamification Tool Design for Reducing Requirements Ambiguity during Elicitation*. http://doi 10.1109/ICCI54321.2022.9756083
14. Husain, M. S. (2021). *Exploiting Artificial Immune System to Optimize Association Rules for Word Sense Disambiguation*. http://doi 10.18201/IJISAE.2021473638
15. Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-Based Software Engineering and Systematic Reviews*.
16. Lucassen, G., & Arendse, B. (2016). *Toward tool Mashups:Comparing and Combinig NLPRE tools*.
17. Maulud, & Dastan, H. (2021). *State of art for semantic analysis of natural language processing*.
18. Moreno, G., Génova José, Fuentes, J. L., & Hurtado Valentín´, O. (2015). *A framework to measure and improve the quality of textual requirements*.

19. Naeem, A., Zeeshan, A., & Ali Shah. (2019). *Analyzing Quality of Software Requirements. A Comparison Study on NLP Tools*.
20. Osama, M., Aya Zaki-Ismail, Mohamed Abdelrazek, John Grundy, & Amani Ibrahim. (2018). *Ambiguous software requirement specification detection: An automated approach*.
21. Petersen, K., S. Vakkalanka, & L. Kuzniarz. (2015). *Guidelines for conducting systematic mapping studies in softwareengineering: An update. Info. Softw. Technol*.
22. Raikar, S. & Cholli N. G. (2021). *An Analysis of Ambiguity Detection Techniques for Software Requirement Specification*.
23. Sabriye, A., Olow, & Jim Ale. (2018). *An approach for detecting syntax and syntactic ambiguity in software requirement specification*.
24. Schmitt X. (2022). *A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate*.
25. Stálhane, T., & Wien, T. (2015). *The DODT tool applied to sub-sea software*.
26. Yadav, A., Patel Aarshil, & Shah Manan. (2021). *A comprehensive review on resolving ambiguities in natural language processing*.
27. Ywang. (2015). *Automatic detection of ambiguous terminology for software requirements*.
28. Zait, F., & Zarour, N. (2019). *Addressing Lexical and Semantic Ambiguity in Natural Language Requirements*.
29. Zhao, L. (2021). *Natural Language Processing for Requirements Engineering: A Systematic Mapping Study*. https://doi.org/10.1145/3444689
30. Zhao, L. (2022). *Classification of Natural Language Processing Techniques for Requirements Engineering*. http://arxiv.org/abs/2204.04282