Automatic Domain-specific Corpora Generation from Wikipedia - A Replication Study

Seniru Ruwanpura*, Cale Morash*, Momin Ali Khan*, Adnan Ahmad*, and Gouri Ginde*
*Dept. of Electrical and Software Engineering, University of Calgary, Canada
Email: {seniru.ruwanpura, cale.morash1, mohammed.alikhan, adnan.ahmad, gouri.deshpande}@ucalgary.ca

***Abstract*—Replication studies help mature our knowledge and attempt to validate the findings of a prior piece of research. However, these studies are still rare in the Requirements Engineering field. Additionally, the rapidly advancing realm of Natural Language Processing (NLP) is creating new opportunities for efficient, machine-assisted workflows application which can bring new perspectives and results to the forefront. Thus, in this paper, we replicate and extend a previous study (baseline), a tool, WikiDoMiner, which automatically generated domain-specific corpora by crawling Wikipedia. In this study, we investigated and executed the implementation of WikiDoMiner (open-sourced code from the original paper) to recreate the results. This allowed us to strengthen the external validity of the original study. We extended the baseline to evaluate additional data sets and generated nuanced results using state-of-the-art NLP techniques such as Bidirectional Encoder Representations from Transformers (BERT). Results showed that due to the growing content in Wikipedia, the corpus generated for the Railways and Networks domains did not precisely match the results from the baseline. However, utilizing the state-of-the-art KeyBERT library from the Huggingface AI community enhanced the results, eventually generating a meaningful corpus compared to the baseline.**

***Index Terms*—Replication study, Requirements analysis, Requirements Engineering, Natural language processing, BERT**

I. Introduction

Replication studies are held as the gold standard for ensuring the reliability of published scientific literature in various domains, including Software Engineering (SE) [1], [2], [3], [4] [5]. For Schmidt [6], "replication experiment to demonstrate that the same findings can be obtained in any other place by any other researcher is proof that the experiment reflects the knowledge that can be separated from the specific circumstances (such as time, place, or persons) under which it was gained". The Software Engineering domain has established the need for replication studies, and such publications have been steadily growing for years [7]. Increasing emphasis on open science research has motivated researchers to make their datasets and complete source code publicly available on persistent data repositories such as Zenodo and GitHub for other researchers to recreate and validate the results to a large extent.

With the advent of rapid technological advancement in Natural Language Processing (NLP) and applied Machine Learning (ML), the Requirements Engineering (RE) domain has benefited extensively due to the textual nature of the prevalent data. In the recent past, Large Language Models (LLM) based approaches that leverage embeddings from pre-trained BERT (Bidirectional Encoder Representations from Transformers) have shown effectiveness in keyphrase extraction [8]. Evaluation of its usefulness remains to be seen in the RE studies [9], where meaningful information identification is crucial. Thus, as part of an undergraduate research project for Software Requirements Engineering, we replicated WikiDoMiner, an automatic domain corpus generation tool [9]. Additionally, we modified the tool further to ascertain improvements to the overall corpus quality. None of the original authors were part of this replication study; we contacted the lead author once to receive information about the original study in the initial stages.

The research questions (RQs) evaluated in this study are as follows:

RQ1: To what extent was the original study replicable?
Rationale: Replicating the results from an open-sourced program and tool to regenerate the results is challenging due to code dependencies and system configurations. Overcoming these challenges, regenerating and evaluating the original study results could help us strengthen the external validity of the results. Additionally, datasets from two additional domains could be analyzed to evaluate the original study further.

RQ2: How do domain-specific corpora generated using LLM based keyword extraction approach: KeyBERT [8] compare with results from WikiDoMiner (baseline)?
Rationale: With the exponential technological ad-

vancements, state-of-the-art NLP techniques have shown tremendous potential in understanding underlying context to a great extent [10]. Thus, enhancing the original study with such a solution could lead to promising improvements to the complete study and domain corpus accumulation strategies.

The contributions made in this paper are as follows:

- Utilizing the source code and the dataset repository from the base paper [9], we executed the source code to recreate the results for corpus generation, thus exploring the external validity of the study.
- We further enhanced the baseline's source code to evaluate various techniques, such as the Cosine Similarity measure and N-gram-based keyword extraction technique.
- State-of-the-art BERT based keyword extraction library was integrated into baseline source code to generate nuanced and accurate domain corpus and compared to the enhanced baseline.

The rest of the paper is organized as follows. Section II provides the overview of the related work and fundamental concepts followed by study design in Section III. Datasets are described in Section IV. Results are explained in Section V followed by threats to validity in Section VI. Finally, the conclusion and future work is explained in Section VII.

II. Fundamentals and Related work

A. Related work

Requirement engineering tasks can be completed efficiently by narrowing the scope to a specific domain. Domain-specific corpora are useful resources for improving the accuracy of automation in RE [11]. In the absence of such a corpus, domain documents from sources such as Wikipedia, books and magazines [12], [13] have been employed by various studies in the field of RE in the past. An automated classification model was introduced for classifying requirements and non-requirements for the automotive domain [14] and for the railway domain [15]. Ezzini et al. introduce TAPHSIR [16], which reviews the use of pronouns in a requirements specification and revises those pronouns that can lead to misunderstandings during the development process. However, most of the existing methods for RE automation require domain-specific knowledge, which limits their application in domains where domain-specific datasets are sparse.

NLP has shown promising results in the recent RE literature and generated significant interest in the RE community, creating a sub-field of its own — NLP4RE [17]. In recent years, BERT [10] has become a state-of-the-art tool for language tasks trained on large text databases, including Wikipedia. Using the attention mechanism [18], BERT can learn about the text's word co-occurrences and semantic contents. Thus, BERT can be fine-tuned for solving downstream tasks for RE. Thus, many of the existing NLP4RE methods must be adapted to BERT models, which have been shown to work better across different tasks. This necessitates adapting the existing solution to improve accuracy and avoiding relying on libraries that will be soon updated or are challenging to maintain. Since the ML community uses Python to enable better usability, it is advantageous to have implementations of NLP4RE in Python.

B. WikiDoMiner (WDM)

WikiDoMiner (WDM) [9] is a tool designed to generate domain-specific corpora by crawling Wikipedia. WDM was designed to generate external knowledge resources specific to the underlying domain of a given requirements specification (RS). This corpus was generated by querying Wikipedia with the extracted keywords and creating a word cloud with the most relevant keywords. WDM was evaluated using the Public Requirements Dataset (PURE) [19] dataset and published in 2022 at the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE' 22).

C. KeyBERT

KeyBERT is a keyword extraction library from Huggingface open source library by the AI community that leverages embeddings from the BERT model [10] to find similar words in a document [8]. BERT is pre-trained on deep bidirectional representations from the unlabeled text by joint conditioning on both left and right contexts in all layers, and research showed that the pre-trained representations reduce the need for many heavily-engineered task-specific architectures. This method contrasts WDM's hand-crafted pre-processing techniques; BERT, trained on large text data, eliminates the need to design pre-processing manually. BERT embedding captures semantic information in the text efficiently and thus has achieved state-of-the-art performance in many NLP tasks. **In this study,** we experiment with a BERT-based KeyBERT library for generating domain-specific corpora generation from Wikipedia.

III. Study designs

Our study design underwent a series of transformations, beginning with a recreation of the baseline

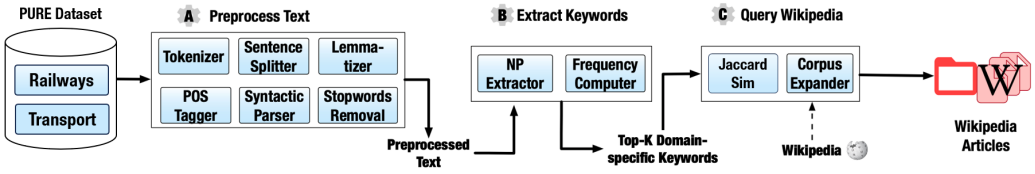

Fig. 1: Original study design: WikiDoMiner (referred to as WDM) [9]

study (WDM) for the corpus generation, followed by a modified version to certain components (En_WDM), and finally, a revamped version using BERT (WDM_BERT).

A. Baseline WDM

Figure 1 shows the original design employed in the baseline WDM. Utilizing RS documents from two domains, namely Railways and Transportation, collected from the PURE [19], the tool pre-processed the document using six techniques (Step A). The first four techniques were to normalize the text, including a tokenizer, a sentence splitter, a lemmatizer to find the canonical form of a word, and a stopwords remover which eliminates irrelevant words utilized in English to communicate fluidly [9]. The other two techniques, a) syntactic parsing, assigns a part-of-speech tag to each token, and b) parser, identifies all units in the text (nouns, verbs, etc.). Using all six techniques, the RS document is put into a state such that keywords can be extracted.

In extracting the keywords, WDM collected all noun phrases in the RS and sorted them based on their frequency of use. This was done through two methods. Firstly, utilizing the WordNet database, common English language words were removed to prevent the eventual corpus from being unrepresentative of the domain from which the RS originated [20]. Then, the term frequency/inverse document frequency (TF/IDF) was calculated for all keywords [21]. Using this score, Top-K keywords were chosen for further processing.

Settings and configurations:

- In this study, the Top-K was set to 50 based on the default value set by the baseline for replicability purposes [9] in study designs III-A, III-B, and III-C.
- Top-K keywords are used to search Wikipedia to build a domain-specific corpus out of the Wikipedia articles. In baseline, matching articles are chosen based on the Jaccard Similarity (1), where

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

J(A,B) is a measure of similarity between two documents *A* and *B*, i.e. the similarity between the keyword and the set of article titles is computed by comparing the intersection and union of their elements [22].

- Threshold value of Jaccard Similarity is set to >0.5 as per parameters of baseline. Then, all articles which only directly matched the extracted keywords with any such score were added to the corpus - this refers to a search depth (how far WDM searches through Wikipedia) of 0. However, since this significantly limits a corpus's usefulness in estimating the frequencies of word co-occurrences and training a domain-specific language model, a depth of 1 was set as the parameter. Since each 0-depth article belongs to a category, a depth of 1 refers to all other articles in the same category being added to the corpus based on the Jaccard score. While there are further sub-categories for depths of 2 and further, since an article depth of 0 and 1 inside the corpus was what the original study used, the WDM baseline did the same.
- Finally, once the corpus was created with the keywords, article titles, and article text, a word cloud was created to highlight the most frequently occurring words in the corpus. This visualized how representative the corpus is of the domain where the RS document originated from.

B. Enhanced WDM (En_WDM)

As a progression to our study, we further enhanced the baseline WDM to replace the Jaccard Similarity score with Cosine. As shown in Figure 2, the querying techniques were revamped to allow for a better similarity score mechanism by replacing Jaccard Similarity with Cosine Similarity and improving pre-processing

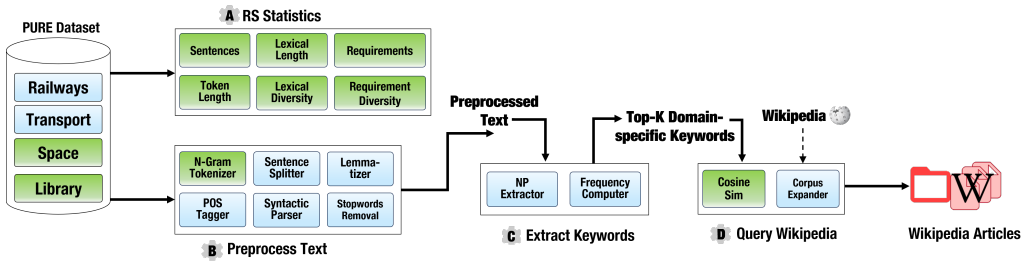

Fig. 2: Enhanced WikiDoMiner (referred to as En_WDM) study design. Components with green color depict new or changed components from WDM (baseline)

mechanism by adding an N-Gram tokenizer. First, in Step A, descriptive statistics were calculated for all three documents in the four chosen domains so that the domains could be better understood. The first document in the domain was chosen for corpus generation, with the other two mainly for future work to judge domain relatedness between the overall results from the first document and the other two.

In Step B, the regular tokenization method only splits text by individual words based on whitespace, punctuation, and other similar metrics and thus cannot account for text context with complex morphology [23]. N-Gram tokenization splits text into fixed-length statements of characters or words called "n-gram", which are either 2 or 3 characters long [24], capture the flow and structure of the given text. While it has disadvantages based on being arbitrary, it is a necessary test to examine whether it produces better results by optimizing pre-processing. Step C is the same as the baseline's Step B (III-A).

Compared to the Jaccard score, the Cosine score effectively generated a corpus based on RS documents with varying lengths and was thus chosen as a replacement in Step D to create the corpus. Additionally, it is chosen due to its connection with the TF/IDF vectorizer [25]. Parallel to Jaccard scores, Cosine scores (in (2)) range from 0 to 1, with a higher score showing higher similarity, thus scores >0.5 were chosen to send articles to the corpus based on the default setting in baseline. Note, that both scores are printed in both these models, but the baseline uses Jaccard whereas En_WDM uses Cosine. These were again on articles of depth 0 and 1.

$$cos(A, B) = \frac{A \cdot B}{||A|| ||B||} \tag{2}$$

C. WDM with BERT (WDM_BERT)

The baseline WDM tool was finally enhanced to incorporate BERT-based keyword extraction libraries [8] to overcome the need for text pre-processing while infusing context-based awareness in the tool.

Figure 3 shows the overall design of this study. Developing on top of the En_WDM, in Step B, KeyBERT was used for word extraction instead of WDM's base code through a lightweight keyword extractor that uses BERT embeddings to extract meaningful keywords and key phrases [10]. It facilitates integration with other NLP libraries and provides a flexible and efficient approach to keyword extraction [8].

In addition to the calculation of Cosine Similarity, Max Sum Distance was also included to increase the number of most similar words to the most similar two-word phrases [26]. As such, 25 unique key phrases were extracted to gather the top $K = 50$ keywords. The top n combinations from these most similar words/phrases are then used to extract the combinations that are least similar to each other through Cosine Similarity, comparing between document and keyword embeddings through the sklearn library [10].

Like WDM and En_WDM, KeyBERT was connected to the pipeline of WDM through updated Wikipedia browsing abilities and word cloud generation. We used the Jaccard Similarity to match the extracted key phrases to the queried Wikipedia articles; however, since in WDM_BERT, the Jaccard score now has two words from the use of key phrases to match articles, the score of 0.5 in the previous methodologies was slightly dropped to the modified threshold of 0.4. The depth of the Wikipedia traversal was set to 0 instead of 1 to generate a more accurate corpus relative to both WDM implementations.

IV. Dataset

In this replication study, we used the PURE [19] dataset, which was also used in the original (baseline) WDM study [9]. The baseline WDM evaluated six Requirement Specification (RS) documents, three each

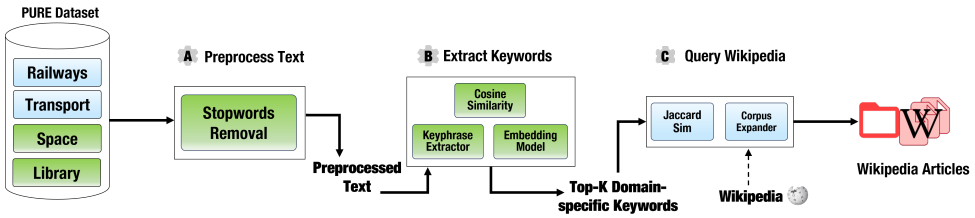

Fig. 3: WikiDoMiner with BERT-based keyword extraction's (referred to as WDM_BERT) study design. Components with green color depict new or changed components from baseline

from two topic domains, namely Railways and Transportation.

The original paper selected one document for corpus generation within each domain. At the same time, the other two would be utilized at the end of the research process to evaluate semantic relatedness to the generated corpus and thus understand how valuable the corpus was against similar RS documents. However, in our study, we focus on evaluating the corpus generation process against each methodology, and modifying the pre-processing and article addition elements of this process. Due to unworkable code for the relatedness aspect in the baseline WDM, that component remains part of our future work.

We included two additional domains with the three RS documents each to enable a meaningful comparison. This was intended to contrast the results from the original PURE RS documents employed in baseline WDM with the new documents selected.

In constructing the two further domains, various RS titles were first examined by the authors of this replication study to find documents of a similar general topic breadth. Instead of basing our selection merely on the informal view of document titles and content, we reviewed and utilized a macro-level descriptive analysis (examining the overall PURE dataset) [19] to narrow down Space and Library tailored RS titles to three documents, each through document structure and length.

Utilizing the original two domains present in the original paper, the informal analysis of RS document titles and content, alongside the macro-descriptive analysis, our final domains were as shown in Table I. The first RS refers to the document used for corpus generation in each domain, whereas the other two are for future work's semantic relatedness.

Before pre-processing and keyword extraction were applied to each domain's RS document, we performed micro-descriptive analysis (analyzed each document in each domain). This was done since pre-processing is based on simplifying sentences and words into tokens in

TABLE I: Various files and related content information for the selected four domains from PURE dataset

Domains	RS file names	#pages
Railways	(RS1) ERTMS, train control	48
	(RS2) EIRENE SYS 15, digital radio standard for railway	138
	(RS3) EIRENE FUN 7, digital radio standard for railway	97
Transportation	(RS4) CTC NETWORK, traffic management networks	32
	(RS5) PONTIS, highway management	82
	(RS6) MDOT, transportation management	56
Space	(RS7) ESA, space data management	54
	(RS8) EVLA BACK, astronomy data management	18
	(RS9) EVLA CORR, astronomy data management	17
Library	(RS10) NLM, digital library management (medicine)	54
	(RS11) LIBRARY, library technology system	18
	(RS12) LIBRARY, library database management system	17

the case of the baseline WDM or with BERT embeddings which pre-train the language word model representation [8]. These tokens were the genesis of the pre-processing and understanding their amount and importance in terms of the keywords that are chosen in the process.
Overall, the descriptive analysis is based on the categories as follows:

- *Sentence Number*: Number of sentences in the running text based on periods
- *Average Sentence Length (Tokens)*: How long each sentence is on average based on the token count (commas and words) [9]
- *Average Sentence Length (Lexical Words)*: How long each sentence is on average based on the unique word count

- *Lexical Diversity*: Ratio of unique words based on stemming (a unique canonical form of a word) to the total amount of words in the text [9]
- *Requirement Number*: Number of overall general requirements in RS document
- *Requirement Diversity*: Ratio of overall unique general requirements to the total amount of sentences in RS

To gather statistical information for our dataset, a python program was written utilizing the NLTK library [27] to apply NLP techniques. This essentially extracted each pre-processing statistic from an individual document. The requirement statistics were counted using a .xls file due to inconsistencies based on structure (accounted for in our macro analysis).

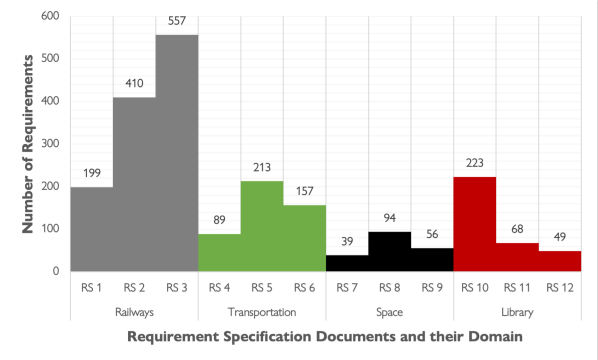

Fig. 4: Number of requirements in each requirements specification (RS) document, grouped by domain

Figure 4 shows that the first file of each domain (the corpus generation file) has relatively variable requirements. This supports the idea that the corpus is related more to that document versus the overall corpus when the generation file's requirement count is higher, with certain keywords not relating to the later documents. For example, RS10, which is focused on a medical library, has much more requirements than the other two RS documents, which means the corpus would be less valuable to a library domain overall and more valuable for medical topics.

In Table II, the data shows the overall usefulness of the documents as RS, illuminating the contrast between requirements versus other content in each domain, thus providing hints towards the pre-processing executed in the three WDM pipelines. Interestingly, results showed higher requirement diversity in Railways versus the other three domains while having the lowest lexical diversity. This high technical nature of the domain could explain eventual results where RS1 does not produce as strong word cloud results representative of the domain, more highly similar technical and less unique regular phrases.

TABLE II: Descriptive analysis of domains used

	Railways	**Transportation**	**Space**	**Library**
#Requirements	1166	459	189	340
#Sentences	2252	1898	589	1142
Avg Token Length	28.51	22.53	24.49	29.89
Avg Lexical Length	21.58	17.56	22.82	23.80
Lexical Diversity	0.11	0.16	0.24	0.21
Requirement Diversity	0.52	0.24	0.32	0.30

V. Results

In this section, we discuss answers to the RQs in detail. Although we conducted extensive experiments for the four chosen domains, only a few select results are presented due to space constraints, with a one-word cloud illustrating key corpus terms for each research question. Complete results can be found at[1].

A. *RQ1: Replication of baseline study*

To answer the first research question, we executed the WDM source code to derive the results for Railways and Transportation domains. Secondly, we used alternative techniques, such as the Cosine Similarity measure and N-gram keyword extraction, to further explore their impact on the results. Specific results explaining these are as follows.

1) RQ1.1 - Extent of replicability of WDM: First, attempting to evaluate the extent to which the original study could be replicated, the original code base was downloaded and executed. This was done to examine whether additional results could be generated using a replica of the study and then comparing these results to Space and Library domains, followed by an overall comparison to an enhanced version of WDM (En_WDM). Several issues were faced while setting up the original WDM tool on our systems. Most notably, an issue with the encoding of the text data came up, which caused errors during the corpus generation process due to unforeseen problems in the source code. Specifically, the error "UnicodeEncodeError: 'charmap' codec cannot encode characters" was observed. This error indicated that the encoding being used could not handle certain characters in the text data. To resolve this issue, the

[1]https://github.com/Protozet/WikiDoMiner

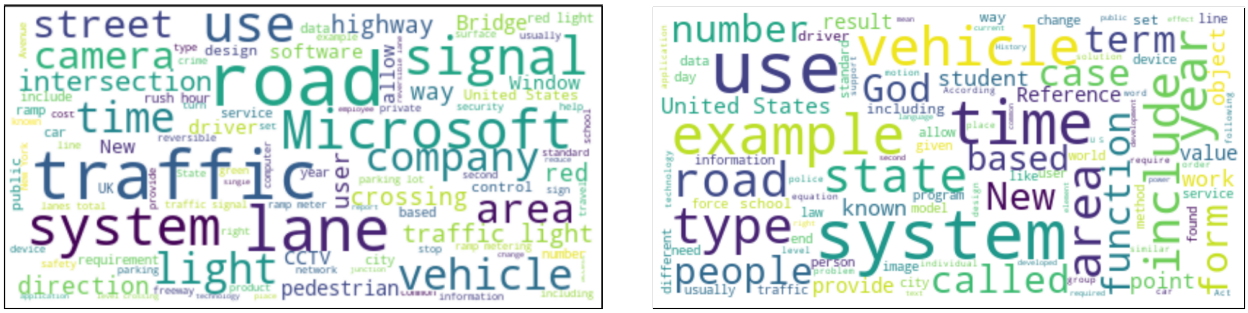

Fig. 5: Word-cloud visualization of domain-specific corpora using original study WDM (Left-hand side) [9] and our Baseline WDM (Right-hand side) for Transportation domain - answering RQ1.1

encoding was changed to UTF-8, a comprehensive encoding that supports a wide range of characters [28].

Once this change was implemented, the corpus-generation process was successful, and the resulting corpora were written out to text files as expected. Such a change highlights the importance of ensuring the encoding is appropriate for processing text data. Failure to do so can result in errors and issues that can impact the accuracy and effectiveness of the corpus generation process. Overall, this issue faced supports the notion that the original study could not be replicated entirely on a practical basis because the original code could not be used as it was; however, attempting to replicate the methodology of the study would still allow an analysis of the extent to which original WDM could be replicated. All results for RQ 1.1 are with the caveat that there is a limited comparison that can be made.

Using K = 50, Railways and Transportation were first tested. A vital sign of replication was that our baseline WDM had Railways produce a total of 689 articles, and in the original study, it produced a corpus of 686 articles [9]. Despite the UTF changes made, this result illustrates high confidence in the similarity of the two programs to crawl Wikipedia. However, changes in the most common words as depicted by the word clouds can be seen, for example, in Figure 5. While this may, at first glance, detract from the reliability of this model, considering that keywords such as vehicle, road, driver, and traffic are still highlighted, the results still show strength in generating domain-specific results.

2) RQ1.2 - Comparing WDM with En_WDM: Cosine uses a TF/IDF vectorizer; however, for this to function, the base program had many errors in the "getCorpus" function, which had to be rewritten and hence adjustments had to be made. A couple of main metrics were utilized to discover the effectiveness of the En_WDM compared to the original. The first is an examination

TABLE III: Comparison of average Jaccard and Cosine scores utilized respectively by baseline WDM and En_WDM, K=50

	Jaccard	Cosine
Railways	0.150	0.194
Transport	0.172	0.220
Space	0.186	0.207
Library	0.171	0.224

of the Jaccard and Cosine scores. As shown in Table III, the Cosine score is always higher than the Jaccard on average. Since En_WDM uses the Cosine while the baseline uses the Jaccard, En_WDM is far more effective in getting articles passed into the corpus along with categories and their articles. Further, the raw data utilized for the mean calculations also showed no instances when a keyword's Jaccard score was higher than the Cosine score. However, due to the use of n-gram, certain corpus's being smaller (Railways and Library) can occur.

Next, as shown in Figure 6, the specificity of the En_WDM corpus creation is visible. While the baseline only generated general terms unspecific to space, En_WDMcan generated phrases like X-Ray, light, and telescope related to space management.

B. RQ2: Comparing En_WDM with WDM_BERT

To evaluate the RQ2, we first restructured the baseline (WDM) implementation and incorporated an interface to utilize the KeyBERT library [8]. As such, we kept the preprocessing minimal since LLM need minimal to no text preprocessing. Figure 3 shows the overall design of this experiment. We evaluated this new tool against four selected domains explained in Section IV. Figure 7 shows the word clouds for the Library domain. Respectively, with the Space Domain in Figure 8.

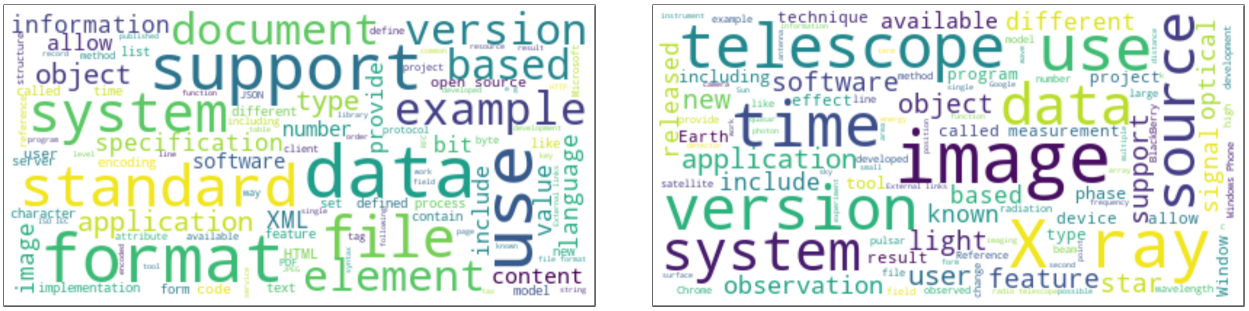

Fig. 6: Word-cloud visualization of domain-specific corpora using baseline WDM (Left-hand side) and Enhanced WDM: En_WDM (Right-hand side) for Space domain - answering RQ1.2

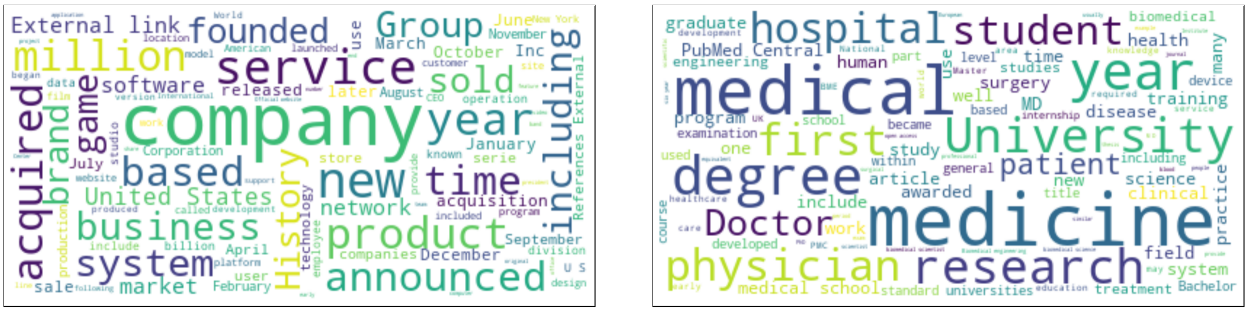

Fig. 7: Word-cloud visualization of domain-specific corpora using En_WDM (Left-hand side) and WDM_BERT (Right-hand side) for Library domain - answering RQ2

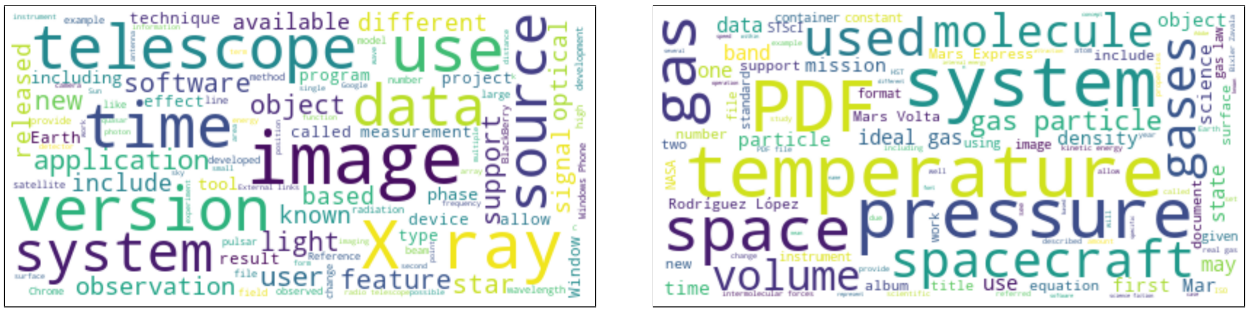

Fig. 8: Word-cloud visualization of domain-specific corpora using En_WDM (Left-hand side) and WDM_BERT (Right-hand side) for Space domain - answering RQ2

Through these word clouds, we show for each domain the main terms that frequently occur in the corpus, which shows that the corpus generated is far more reflective of the RS document content. This is visible in the Library domain as well. RS10 (which generated the corpus) (Figure 7) mostly contained PubMed documents; hence we could see the occurrence of terminologies such as medicine, hospital, physician, etc. in the WDM_BERT generated results compared to random unrelated content depicted in word cloud from En_WDM (and indeed baseline). Further, in Table IV, the quantitative edge of WDM_BERT is exhibited through the average Jaccard score. In each domain, though the amount of articles produced is much lower than En_WDM (and indeed the baseline) due to a depth of 0, the Jaccard score is significantly higher. In conjunction with the more accurate word cloud results, this suggests that the final method is the best corpus generating methodology with the more significant similarity between keywords and articles, alongside overall better article results as shown

in the word cloud.

TABLE IV: Comparison of average Jaccard and Cosine scores respectively printed and used by En_WDM, and average Jaccard score used by WDM_BERT

	Jaccard En_WDM	**Cosine En_WDM**	**Jaccard WDM_BERT**
Railways	0.149	0.194	0.25
Transport	0.173	0.220	0.517
Space	0.181	0.207	0.537
Library	0.169	0.224	0.589

VI. Threats to validity

A. External threats:

We do not fine-tune the BERT model for the domain-specific data. However, base models of BERT are trained on Wikipedia data, which mitigates this threat. ML models often have an implicit bias from the training data, which may make the generated corpus biased. However, further study will be required to study the bias for RE which is beyond the scope of this work.

B. Internal threats:

The selection of different RSs within the same domain might yield different results. However, the lexical diversity of the pool is not significant (Table II); thus, this threat is taken care of. Increasing the number of keywords used to query Wikipedia could generate different results. However, we have conducted various tests during this study which did not yield a significant variation in the results. Hence, we anticipate its lower impact on the overall results. Increasing the similarity score threshold value could affect the overall number of documents fetched from Wikipedia. However, we envision doing an extensive ablation study as part of future work, and as 0.5 is the accepted value with both similarity scores, this weakness is likely mild.

The baseline tool was set up to explore higher depths than 1; however, in this replication study, a depth of only 1 was selected for both our implementation of the baseline and En_WDM to balance the creation of a useful corpus while not having one so large with extraneous information. Further with WDM_BERT, unlike the baseline study, we restricted our search traversal depth to level 0 while skipping subsequent levels in order due to a correctly hypothesized ability to create a better corpus with more directly relevant articles. Outside of the better word clouds, this was further justified with WDM_BERT only producing 2 articles with a Jaccard score of 0.25 whereas the original study produced 25 articles [9] . Thus, overall less articles are produced in general but this allows for greater result accuracy. Regardless, since this might have implications for the results; we intend to evaluate this in our future work.

A considerable threat is that the final part of the original study (semantic relatedness) was not conducted in this study, due to unworkable code. Due to previously mentioned tuple inconsistencies and encoding problems in the original code, the semantic relatedness could not be computed, with modifications to the code requiring additional efforts. Unlike changes to the corpus generation process, since semantic relatedness compares different datasets, it was decided not to pursue this element for greater accuracy and reliability in this study. Thus, the overall relationship between the RS used for corpus generation could not be measured against the other RS's used in the descriptive analysis similar to the original study. Due to this, it is not known whether certain results represent the test RS more than the overall domain. As this was not conducted due to issues with the original code, this has been envisioned as part of our future work to uncover an answer, where benchmarking against other relatedness methodologies will be performed.

Finally, with some of the word clouds, notably Railways, the results were not fully representative of the domain (RS document); however, since the quantitative results show similar Cosine and Jaccard scores and the hypothesized impact of lexical diversity versus requirement diversity was shown in the descriptive analysis, this threat likely has understandable reasoning. Further, the average Jaccard score through WDM_BERT is significantly lower than the otherwise high scores for domains, suggesting an issue with the RS and Domain itself.

VII. Conclusion and Future Work

For most of the characteristics evaluated with this study, the results indicate that replication study followed by enhancements to the study design improved the overall results primarily based on better Cosine scores than Jaccard on average and generally better word clouds. Additionally, the outcomes were twofold. Firstly it enabled novice and budding researchers to learn nuances of research on safe ground. Secondly, it enabled the advancement of the research in the corpora generation automation domain using state-of-the-art methods mentioned as future work in the original (WDM) publication.

In particular, regarding the replicability of the original study, some challenges had to be tackled to get the tool working on our systems due to various program and system dependencies along with inconsistencies with array iterations. In the process, much-needed emphasis

was levied on learning the complete implementation and design of the study, and results had to be generated with the caveat that there was no way to fully test the original program. Our methodology provides a new, confirmed way of running this corpus generation process.

Regarding the quality of the recreated results, due to the ever-growing content in Wikipedia, the replication of the original study only partially matched the recreated study. However, results from the original study appeared to be a subset of the replication study. Exploring this in detail is part of our future work. The results also confirm our first expectations that utilizing state-of-the-art NLP approaches such as BERT improved the quality of the overall result, which is evident in the word clouds generated for the additional two domains (Space and Library) explored in this replication study. Our results thus strengthen our confidence in the general benefits of automation of domain corpora generation from publicly available repositories such as Wikipedia. We would also like to experiment with new large language models (LLMs) in future work.

ACKNOWLEDGEMENT

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC Discovery Grant RGPIN-2023-03365

REFERENCES

[1] J. C. Carver, "Towards reporting guidelines for experimental replications: A proposal," in *1st inter. workshop on replication in empirical software engineering*, vol. 1, pp. 1–4, 2010.

[2] F. Shull, V. Basili, *et al.*, "Replicating software engineering experiments: addressing the tacit knowledge problem," in *Proc. inter. symposium on empirical software engineering*, pp. 7–16, IEEE, 2002.

[3] F. J. Shull, J. C. Carver, *et al.*, "The role of replications in empirical software engineering," *Empirical software engineering*, vol. 13, pp. 211–218, 2008.

[4] B. Penzenstadler, J. Eckhardt, *et al.*, "Two replication studies for evaluating artefact models in re: results and lessons learnt," in *2013 3rd Inter. Workshop on Replication in Empirical Software Engineering Research*, pp. 66–75, IEEE, 2013.

[5] A. Yates and M. Unterkalmsteiner, "Replicating relevance-ranked synonym discovery in a new language and domain," in *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pp. 429–442, Springer, 2019.

[6] S. Schmidt, "Shall we really do it again? the powerful concept of replication is neglected in the social sciences," *Review of general psychology*, vol. 13, no. 2, pp. 90–100, 2009.

[7] F. Q. Da Silva, M. Suassuna, *et al.*, "x," *Empirical Soft. Engineering*, vol. 19, pp. 501–557, 2014.

[8] "Hf keybert - a hugging face space by hellisotherpeople." https://huggingface.co/spaces/Hellisotherpeople/HF-KeyBERT. (Accessed on 06/12/2023).

[9] S. Ezzini, S. Abualhaija, *et al.*, "Wikidominer: wikipedia domain-specific miner," in *Proc. 30th ACM Joint European Software Engineering (SE) Conf. & Symposium on the Foundations of SE*, pp. 1706–1710, 2022.

[10] J. Devlin *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[11] S. Ezzini, S. Abualhaija, C. Arora, *et al.*, "Using domain-specific corpora for improved handling of ambiguity in requirements," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 1485–1497, IEEE, 2021.

[12] G. Cui, Q. Lu, *et al.*, "Corpus exploitation from wikipedia for ontology construction.," in *LREC*, 2008.

[13] A. Ferrari, B. Donati, *et al.*, "Detecting domain-specific ambiguities: an nlp approach based on wikipedia crawling and word embeddings," in *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pp. 393–399, IEEE, 2017.

[14] J. P. Winkler and A. Vogelsang, "Using tools to assist identification of non-requirements in requirements specifications–a controlled experiment," in *Requirements Engineering: Foundation for Software Quality: 24th Inter. Working Conf., REFSQ 2018, Proc. 24*, pp. 57–71, Springer, 2018.

[15] A. Ferrari, G. Gori, *et al.*, "Detecting requirements defects with nlp patterns: an industrial experience in the railway domain," *Empirical Software Engineering*, vol. 23, pp. 3684 – 3733, 2018.

[16] S. Ezzini, S. Abualhaija, *et al.*, "Taphsir: towards anaphoric ambiguity detection and resolution in requirements," in *Proc. of the 30th ACM Joint European Software Engineering Conf. and Symposium on the Foundations of Software Engineering*, pp. 1677–1681, 2022.

[17] L. Zhao, W. Alhoshan, A. Ferrari, *et al.*, "Natural language processing (nlp) for requirements engineering: A systematic mapping study," 2020.

[18] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," *arXiv preprint arXiv:1906.04341*, 2019.

[19] A. Ferrari, G. O. Spagnolo, *et al.*, "Pure: A dataset of public requirements documents," in *2017 IEEE 25th Inter. Requirements Engineering Conference (RE)*, pp. 502–505, 2017.

[20] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, p. 39–41, nov 1995.

[21] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proc. of the first instructional conference on machine learning*, vol. 242, pp. 29–48, Citeseer, 2003.

[22] S. Niwattanakul, J. Singthongchai, *et al.*, "Using of jaccard coefficient for keywords similarity," in *Proc. of the international multiconference of engineers and computer scientists*, vol. 1, pp. 380–384, 2013.

[23] S. Kannan, V. Gurusamy, *et al.*, "Preprocessing techniques for text mining," *Inter. Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2014.

[24] P. McNamee and J. Mayfield, "Character n-gram tokenization for european language text retrieval," *Information retrieval*, vol. 7, pp. 73–97, 2004.

[25] R. M. Jones, "Evaluation of the effectiveness of cosine similarity in predicting relevance between paired citing and cited sentences.," 2009.

[26] N. Giarelis, N. Kanakaris, *et al.*, "A comparative assessment of state-of-the-art methods for multilingual unsupervised keyphrase extraction," in *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 Inter. Conf., AIAI 2021, Hersonissos, Crete, Greece*, pp. 635–645, Springer, 2021.

[27] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[28] F. Yergeau, "UTF-8, a transformation format of ISO 10646," Tech. Rep. 3629, Nov. 2003.