Finding Similar Repositories based on their Documentation

Alexandru Catalin Turcu a.c.turcu-1@student.tudelft.nl Dr.Ing. Sebastian Proksch, Shujun Huang

1. Background

- In 2022 over 52 million repositories have been created on GitHub alone [1], making the process of finding similar repositories harder.
- These repositories can share valuable insights through the available code and documentation, crucial firsthand experience for new-comers.
- Process of identifying relevant projects to become role models can become overwhelming due to their sheer amount.
- Tools that compare repositories often do not consider the documentation in the process of finding similarities.
- **Idea**: Compare the documentation of repositories and evaluate their content similarity. We consider comments from source files, as well as Readme and Wiki files.
- Two **repositories are similar** when their goal is to complete the same task, regardless of the technology, or when the end goal is different, but they have a common methodology to get there.

2. Research Question

How similar are GitHub projects that share attributes on the documentation side?

- **What segments of each documentation dimension are the most relevant for finding similarities?**
  - **Which branch (dimension) or combination outputs the best results?**
  - **Should the lack of documentation make two projects similar or not?**

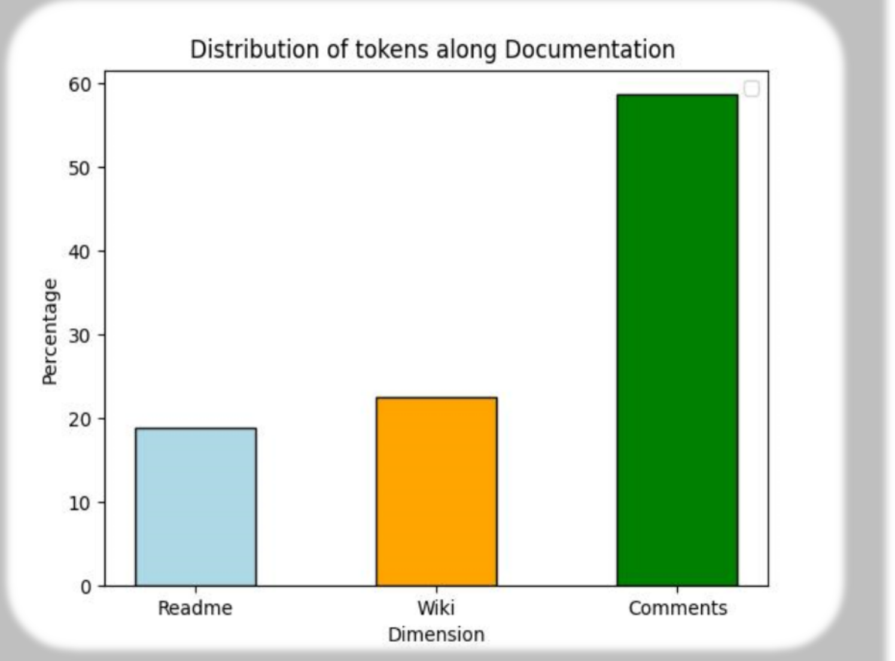

Fig. 2. Documentation distribution in selected Repositories

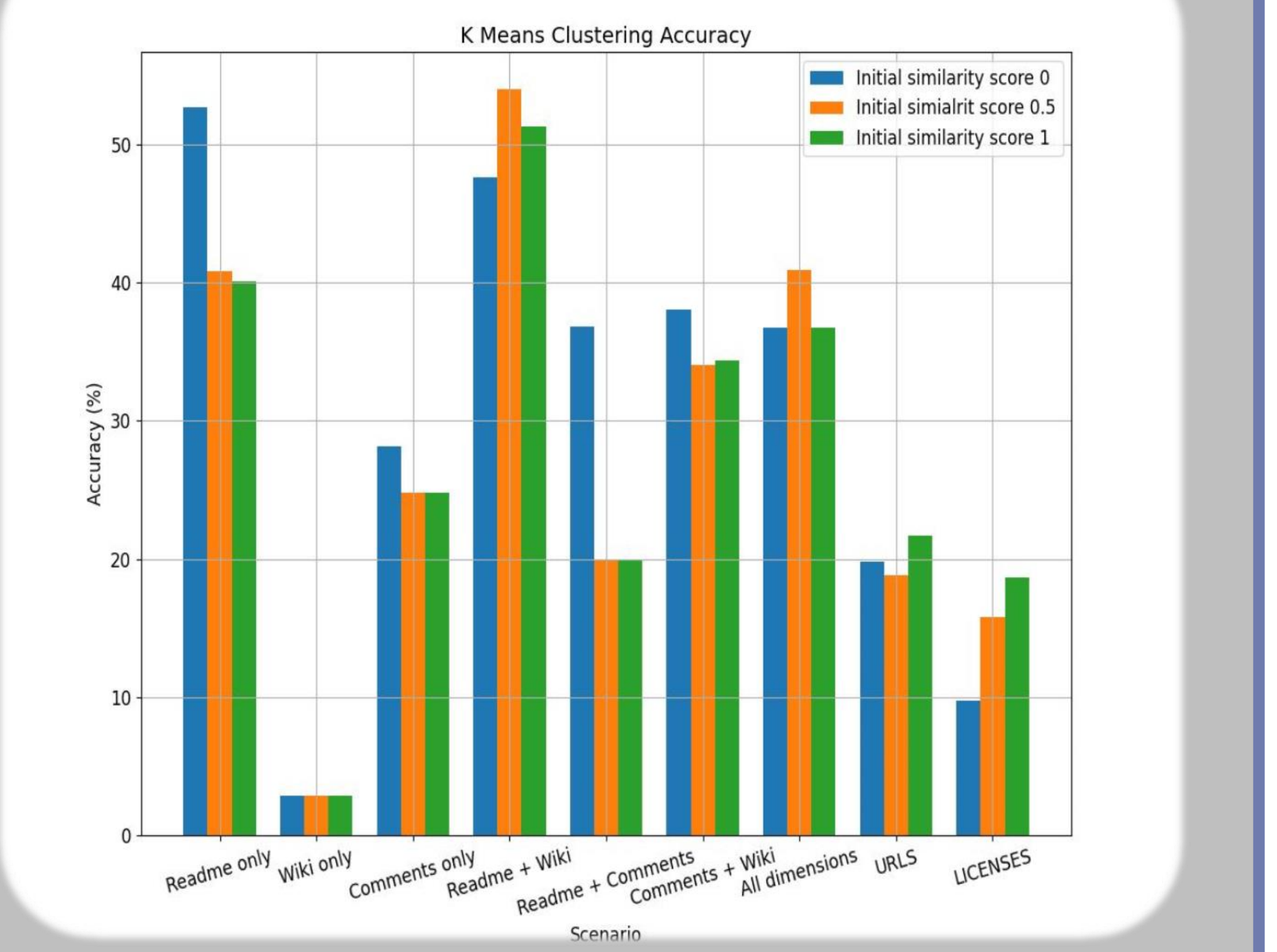

Fig. 3. Example of K-means Clustering accuracy over 9 scenarios with all the cases of handling missing documentation during the analysis experiment

3. Methodology

- Create two datasets, one for each experiment: analysis and validation. The first one is a manually picked and labeled repositories list, while the second uses the dataset used in CrossSim experiments [2].
- From each repository, extract three key aspects: meaningful processed stemmed words, URLs and Licenses.
- Consider 7 scenarios of Readme, Wiki and Comments combinations, and 2 separate ones with URLs and Licenses.
- For the former, vectorize the data with TF-IDF and use cosine similarity to gather a similarity score between repositories. Dimensionality reduction is obtained by using SVD.
- For the latter, save entries into lists of unique elements and find the similarity using Jaccard distance.
- For missing dimensions, consider 3 cases of handling them: 0, 0.5 and 1 as initial similarity score. This is a common occurrence for Wiki pages.
- The first experiment measures the accuracy of clustering of the repositories, while the second experiment considers the top 10 similarity wise repository pairs and manual evaluation is performed.

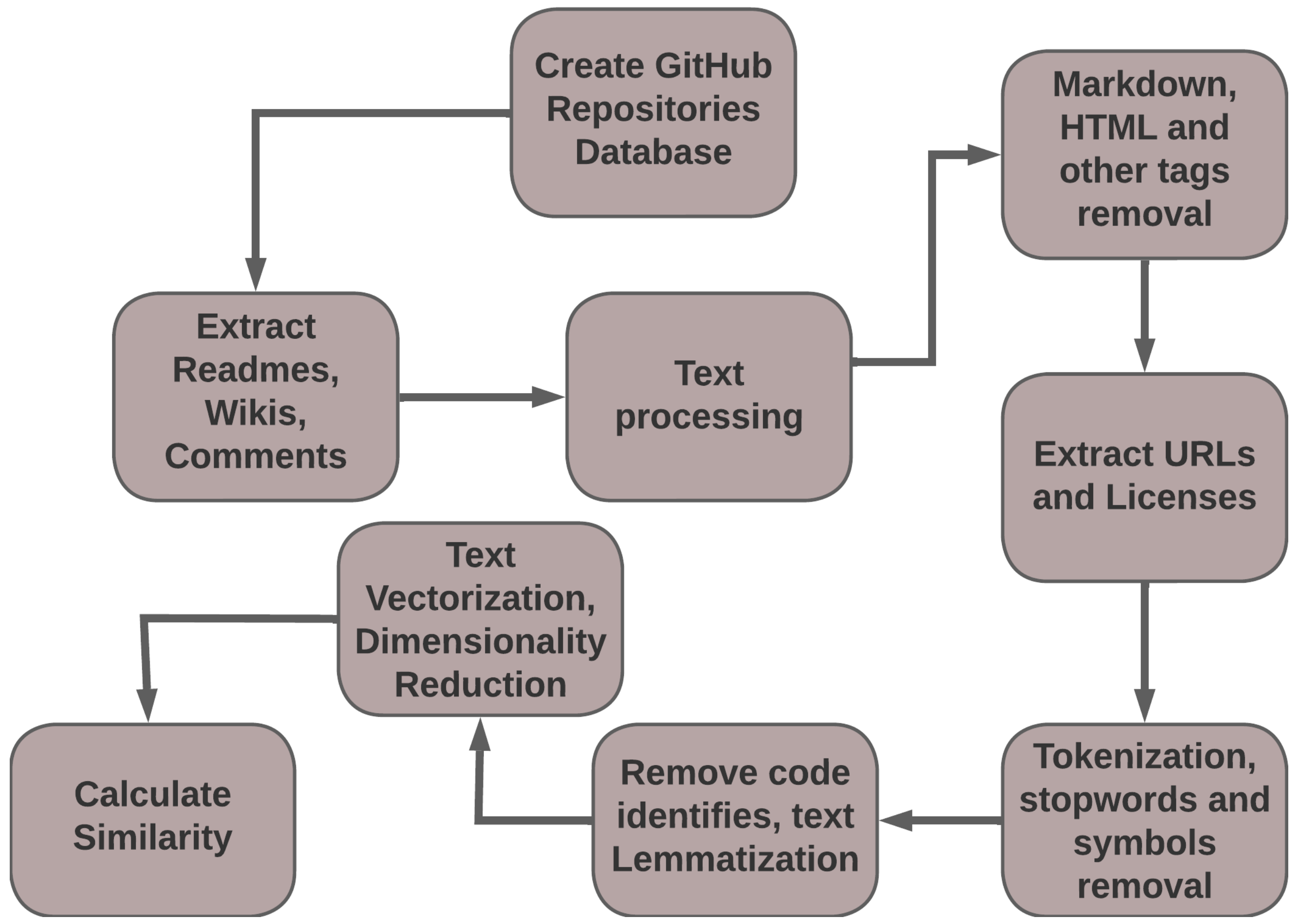
Create GitHub Repositories Database
Extract Readmes, Wikis, Comments
Text processing
Markdown, HTML and other tags removal
Extract URLs and Licenses
Tokenization, stopwords and symbols removal
Remove code identifies, text Lemmatization
Text Vectorization, Dimensionality Reduction
Calculate Similarity

Fig. 1. Methodology overview

4. Results and Observations

- The Comments dimension dominates the other, as shown in Figure 2, which contains the distribution of the dimensions.
- URLs and Licenses performed badly in comparison to the other scenarios.
- We observed similar results when clustering repositories regardless of the missing documentation case. However, giving extreme scores (0 or 1) increases the number of False Positives and False Negatives when comparing repositories.
- The analysis experiments (Figure 3) showed that 'Wiki only' scenario has a low accuracy level, mostly due to missing documentation. This result is confirmed during the validation process. In contrast, 'Readme only' similarity might be misleading, as observed in the validation experiment.
- Best performing scenario is Readme + Wiki, followed by Readme + Wiki + Comments (All dimensions).
- **Documentation comparison represents a valid approach to explore a similarity relationship between repositories.**

5. Limitations & Future Improvements

- Further investigate the URLs and Licenses segments and improve their methodologies.
- Additional optimization for comments extraction and filtration.
- Further evaluation, including user evaluation of the repositories deemed similar.
- Compare the behavior of current tools that consider the comparison of Readmes to adapted versions using all Documentation dimensions.

References

[1] M. Woodward, "Octoverse 2022: 10 years of tracking open source," Nov 2022
[2] P. T. Nguyen, J. D. Rocco, R. Rubei, and D. D. Ruscio, "Crosssim: Exploiting mutual relationships to detect similar oss projects," in 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 388–395, Aug 2018