Datareq: um dataset de requisitos em português

John V. Santos[1], **Antônio K. C. Primo**[2], **Gisele P. Ribeiro**[1], **Diego N. Mariano**[2]

[1]Centro de Tecnologia – Universidade Federal do Ceará (UFC)
– Fortaleza – CE – Brazil

[2]Centro de Ciências – Universidade Federal do Ceará (UFC)
– Fortaleza – CE – Brazil

Abstract. *This work introduces Datareq, a new Portuguese dataset for requirements engineering, consisting of 1,502 software requirements extracted from 8 public notices available online. Each requirement was manually labeled with three attributes: type (functional or non-functional), category, and language (system). Datareq aims to provide data for training supervised learning models and natural language processing (NLP), fostering advancements in tasks such as requirements classification. To assess the utility of Datareq, Support Vector Machine (SVM) models with a linear kernel and Naive Bayes models (Multinomial and Bernoulli) were trained for three classification tasks: functionality, single-category classification, and multi-category classification (Top-2). The results demonstrated the feasibility of Datareq for these tasks, with SVM achieving the best performance, reaching 83% accuracy in functionality classification, 48.5% in single-category classification, and 75% in multi-category classification.*

Resumo. *Este trabalho apresenta o Datareq, um novo conjunto de dados em português para engenharia de requisitos, composto por 1502 requisitos de software extraídos de 8 editais públicos disponíveis na internet. Cada requisito foi rotulado manualmente com três atributos: tipo (funcional ou não funcional), categoria e linguagem (sistema). O Datareq tem como objetivo fornecer dados para o treinamento de modelos de aprendizado supervisionado e processamento de linguagem natural (PLN), visando avanços em tarefas como classificação de requisitos. Para avaliar a utilidade do Datareq, foram treinados modelos de Support Vector Machine (SVM) com kernel linear e Naive Bayes (Multinomial e Bernoulli) para três tarefas de classificação: funcionalidade, classificação unitária de categorias e classificação múltipla de categorias (Top-2). Os resultados demonstraram a viabilidade do Datareq para essas tarefas, com o SVM apresentando melhor desempenho, atingindo 83% de acurácia na classificação de funcionalidade, 48,5% na classificação unitária e 75% na classificação múltipla.*

1. Introdução

Os requisitos desempenham um papel fundamental no desenvolvimento de software. A análise de requisitos, embora essencial, pode ser uma tarefa complexa, devido a fatores como a grande quantidade de requisitos presentes em um único documento de especificação de software (em inglês, Software Requirements Specification ou SRS), a presença de ambiguidades e a variedade de técnicas e padrões empregados na sua elaboração.

Este trabalho apresenta o Datareq, um conjunto de requisitos em português brasileiro. O Datareq consiste em 1502 requisitos extraídos de 8 editais disponibilizados publicamente na internet. Cada requisito é caracterizado por três atributos, rotulados manualmente: funcionalidade (funcional ou não funcional), tipo (de sistema ou de usuário, quanto a sua linguagem técnica ou usual) e duas categorias (Categoria 1 e Categoria 2), abrangendo um conjunto de 13 categorias que definem o tipo do requisito e que podem ser visualizados na **Tabela 4.4**. Do total de requisitos, 1200 são funcionais e 302 são não funcionais.

2. Trabalhos relacionados

Em [A. Ferrari and Gnesi 2017], [Senger 2022] e [Yucalar 2023], foram desenvolvidos datasets de requisitos de software em linguagem natural extraídos de documentos públicos de requisitos da web, disponíveis em inglês e turco, respectivamente. Os três trabalhos realizaram a rotulação manual dos dados para viabilizar o aprendizado supervisionado. Os trabalhos de Ferrari e Yucalar estabeleceram objetivos mais gerais, como classificação de requisitos e atividades com linguagem natural, enquanto Senger se direcionou a previsão de riscos. Os estudos de [T. Hey and Tichy 2020], Yucalar e [Abdur Rahman 2023] investigaram diferentes algoritmos para a classificação de requisitos. Enquanto Hey e Yucalar utilizaram modelos transformadores baseados no BERT, Yucalar também explorou uma variedade de algoritmos tradicionais de machine learning. Em relação aos objetivos de classificação, Hey e Yucalar focaram na diferenciação entre requisitos funcionais e não-funcionais, ao passo que Abdur priorizou a classificação de requisitos não-funcionais, avaliando o desempenho nessa categoria específica.

Abdur e [G. Y. Quba and AlZu'bi 2021] utilizaram o dataset recorrente para classificação de requisitos, PROMISE. Em Quba, foi desenvolvido um estudo que usa algoritmos de machine learning focado em precisão no momento da classificação de requisitos em uma base de dados chamada PROMISE_exp. Em [Lu and Liang 2017], o dataset para classificação de requisitos é a avaliação de usuários das lojas de aplicativos usando algoritmos de machine learning como Naive Bayes, J48 e Bagging. O artigo [de Araújo and Marcacini 2021] trata dos mesmo tipo de dados que o anterior mas usando LM BERT.

3. Metodologia

O Datareq é um conjunto de requisitos de software em português brasileiro, composto por 1502 requisitos extraídos de 8 editais públicos disponibilizados na internet. A coleta dos dados priorizou documentos em formato PDF acessíveis em páginas cujas URLs terminavam em .ORG ou .GOV. Os editais governamentais referentes a licitações públicas para desenvolvimento de software, conta como maior parte dos editais. Os requisitos foram extraídos manualmente dos PDFs e integrados ao Datareq sem quase nenhuma alteração no texto original, preservando a redação presente nos documentos fonte. Das oito fontes de requisitos, seis foram publicadas entre 2018 e 2024, e uma em 2011. A data de origem da oitava fonte, no entanto, não pôde ser determinada. Os dados utilizados neste estudo, incluindo a planilha com os rótulos dos requisitos, estão disponíveis em um repositório público no GitHub.

Após a extração, os requisitos foram rotulados manualmente com quatro atributos: funcionalidade, tipo, Categoria 1 e Categoria 2. O atributo funcionalidade classi-

Categoria	Descrição
Operacional	Relaciona-se com suporte a sistemas operacionais, redes ou tecnologias.
Usabilidade	Descreve funções do sistema que o usuário pode interagir.
Segurança	Requisitos de gerenciamento de permissões e proteção contra atacantes.
Acessibilidade	Garante que o software seja acessível a pessoas com diferentes habilidades.
Legal	Abrange requisitos relacionados a conformidade legal ou regulatória.
Portabilidade	Diz respeito à capacidade de transferir o software para diferentes ambientes.
Escalabilidade	Habilidade do software de crescer em capacidade sem perda de desempenho.
Tolerância a falha	Requisitos para que o sistema continue funcionando mesmo em caso de falhas.
Manutenibilidade	Refere-se à manutenção do sistema, como correções e atualizações.
Performance	Relaciona-se ao desempenho, como tempos de resposta do sistema.
Ciclo de vida	Ciclo de vida do software, como manutenção, descontinuação ou atualização.
Casos de uso	Requisitos que descrevem interações específicas entre usuários e o sistema.
Conformidade	São requisitos relacionados a padrões de código ou estilo.

Table 1. Rótulos dos requisitos do Datareq e suas descrições

fica os requisitos como funcionais (descrevendo *o que* o sistema faz) ou não funcionais (descrevendo *como* o sistema deve se comportar, como desempenho, segurança e usabilidade). O atributo tipo, baseado na linguagem empregada no requisito, classifica-os como requisitos de sistema (com linguagem mais técnica e detalhada, voltados para desenvolvedores) ou requisitos de usuário (com linguagem mais compreensível para usuários finais e stakeholders). Essas características de requisitos, que distinguem entre *o quê* e o *como* e entre a linguagem técnica e a orientada ao usuário, são amplamente discutidas na literatura de engenharia de requisitos, como em [SOMMERVILLE 2011].

Além da funcionalidade e do tipo, os requisitos foram rotulados manualmente quanto à sua categoria. Para lidar com possíveis ambiguidades e requisitos que pudessem se enquadrar em mais de uma categoria, cada requisito foi rotulado com até duas categorias distintas (por exemplo, um requisito poderia ser rotulado como *Segurança* e *Performance*). Foram consideradas treze categorias amplamente utilizadas na descrição de requisitos de software, como exemplificado em [Souvik 2020] e [Shukla 2024]. Essas categorias abrangem diferentes aspectos dos requisitos, incluindo: Operacional, Usabilidade, Segurança, Casos de Uso, Acessibilidade, Portabilidade, Manutenibilidade, Tolerância a Falhas, Ciclo de Vida, Legal, Escalabilidade e Conformidade. A **Tabela 4.4** apresenta uma descrição detalhada de cada categoria, fornecendo definições para melhor compreensão.

4. Resultados

4.1. Avaliação do dataset

O Datareq é um conjunto de dados composto por 1502 requisitos funcionais e não funcionais, cada um rotulado com até dois rótulos em um conjunto de 13 categorias. Seu objetivo é fornecer dados para o treinamento de modelos supervisionados e de processamento de linguagem natural (PLN). A **Figura 1** exibe a distribuição de categorias I.

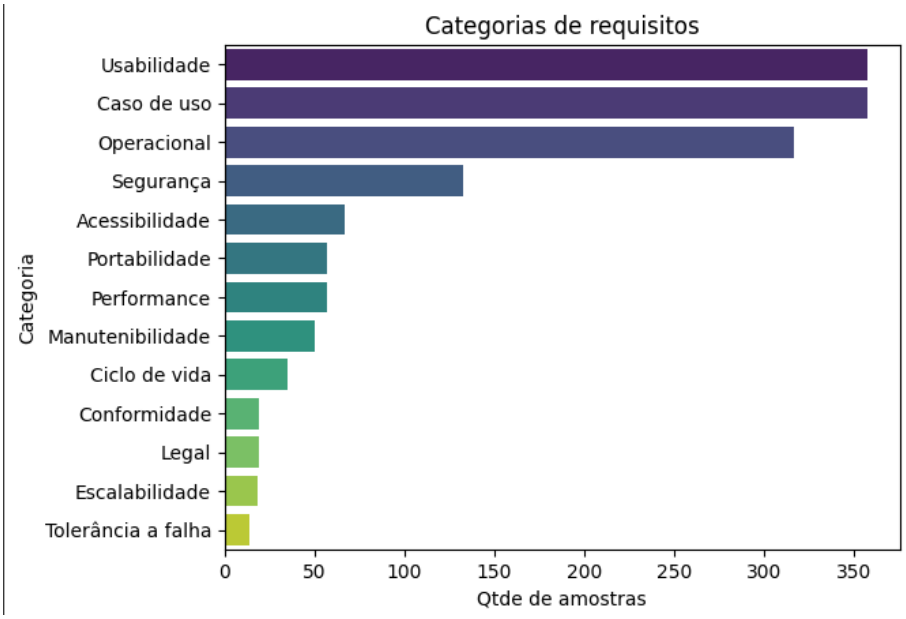

Figure 1. Distribuição das categorias primárias dos requisitos no Datareq

No entanto, o Datareq apresenta vieses significativos. Uma parcela considerável dos requisitos extraídos dos editais (75%) foi classificada como funcional. Uma visão da distribuição de funcionalidades de requisitos pode ser observada na **Figura 2**. Adicionalmente, todos os requisitos foram rotulados como requisitos de sistema, caracterizados por linguagem técnica. Em trabalhos futuros, mais fontes distintas de requisitos serão recolhidos, a fim de obter linguagens mais diversas. Outro viés notável reside na distribuição das categorias, conforme ilustrado na **Figura 1** . Observa-se uma predominância acentuada de requisitos de casos de uso e de usabilidade, seguidos por requisitos operacionais. Devido à presença desses vieses, a **Seção 4.5** discutirá alternativas para mitigar seus efeitos, incluindo técnicas para aumento da variância dos dados.

4.2. Avaliação da classificação

Neste trabalho, o Datareq foi utilizado para o treinamento de dois modelos supervisionados de classificação: Support Vector Machine (SVM) e Naive Bayes. A implementação de ambos os modelos foi realizada utilizando a biblioteca scikit-learn (sklearn) em Python. A escolha desses algoritmos se justifica pela simplicidade e popularidade de ambos. Além disso, o Naive Bayes demonstra bom desempenho em tarefas de classificação textual, como a detecção de spam em e-mails. Assim, como a descrição dos requisitos é uma variável fundamental da classificação, esse algoritmo foi escolhido.

Para cada algoritmo, foram realizados três tipos de classificação. Em cada tipo de classificação foi utilizada validação cruzada com quatro partes a fim de obter uma melhor generalização para o modelo. Nesse contexto, o conjunto de treinamento utilizou 75% dos dados, enquanto o conjunto de testes utilizou 25%

1. **Classificação de Funcionalidade:** Os requisitos foram classificados como funcionais ou não funcionais, utilizando a descrição do requisito como principal feature (característica).

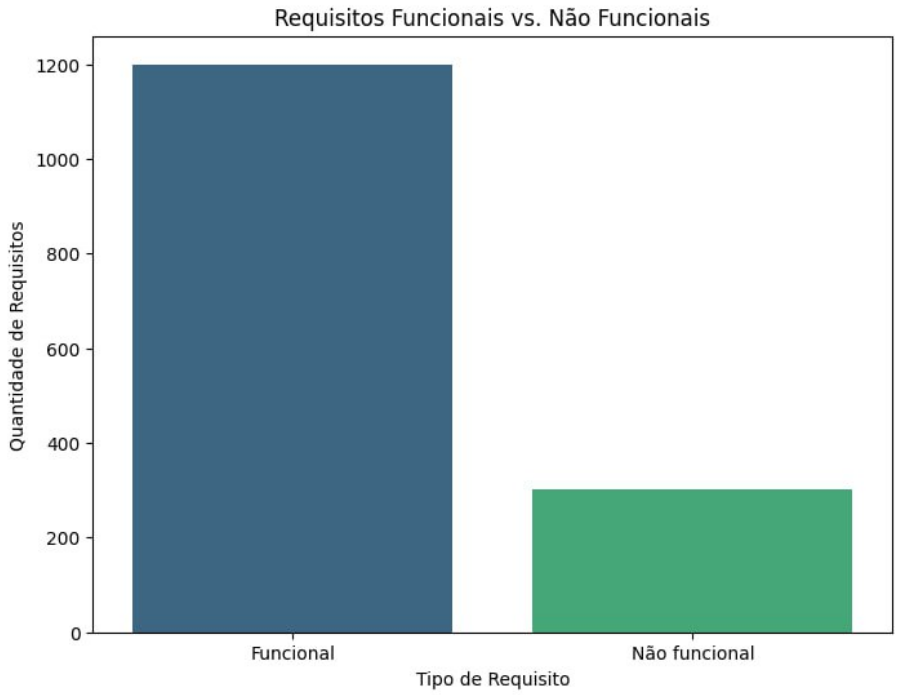

Figure 2. Divisão entre requisitos funcionais e não funcionais no Datareq

2. **Classificação de Categorias (com Funcionalidade):** As categorias dos requisitos foram classificadas com base tanto na descrição do requisito quanto na sua funcionalidade (funcional ou não funcional).

3. **Classificação de Múltiplas Categorias (Top-2):** Devido à ambiguidade presente em algumas descrições de requisitos, este tipo de classificação buscou identificar as duas categorias mais prováveis para cada requisito.

Esta abordagem permitiu explorar diferentes aspectos da rotulação do Datareq e avaliar o desempenho dos modelos em tarefas de classificação com diferentes níveis de complexidade. Para essa avaliação, consideramos tanto a acurácia dos modelos, quanto a métrica F1-score, usada para avaliação de modelos desbalanceados. Dois outro modelos foram avaliados com técnicas de diminuição de viés sobre os dados, comentamos mais sobre eles na **Seção 4.5**.

4.3. Desempenho do SVM

O Support Vector Machine (SVM) apresentou o melhor desempenho entre os modelos avaliados, sendo treinado com o Datareq nos três tipos de classificação propostos. Em todos os casos, utilizou-se o kernel linear. O hiperparâmetro C foi definido empiricamente como 3, resultando nos melhores resultados de acurácia. Os resultados de acurácia obtidos pelo SVM foram:

Tipo de classificação	Acurácia	F1-score
Funcionalidade	83%	80%
Categorias (simples)	48,5%	46.8%
Categorias (Top-2)	75%	NA

4.4. Desempenho do Naive Bayes

O Naive Bayes apresentou desempenho inferior ao SVM em duas das três tarefas de classificação. As versões Multinomial e Bernoulli do Naive Bayes foram avaliadas, apresentando diferenças de acurácia insignificantes entre si em todos os casos. Os resultados de acurácia obtidos foram:

Tipo de classificação	Acurácia	F1-score
Funcionalidade	83%	80%
Categorias (simples)	44%	38%
Categorias (Top-2)	65%	NA

4.5. O viés do Datareq

Apesar dos resultados de classificação obtidos com o Datareq, observamos a presença de um viés significativo no conjunto de dados, o que impacta o desempenho dos modelos. A **Figura 3** ilustra a matriz de confusão do SVM para a classificação unitária de categorias. Nela, é evidente que as categorias mais frequentes, como requisitos Operacionais, de Usabilidade e Casos de uso, apresentam um número considerável de falsos positivos, sugerindo um possível overfitting do modelo para essas categorias.

Diferentes modelos de aprendizado de máquina possuem diferentes susceptibilidades ao overfitting. No entanto, no caso do Datareq, a presença de um viés acentuado na distribuição das categorias contribui significativamente para esse fenômeno. O modelo Naive Bayes também apresentou indícios de overfitting, embora em menor grau se comparado ao SVM. [Faceli 2021] propôs um método para avaliação de modelos de alto viés. O modelo se mostra funcional se ele apresenta acurácia superior a um modelo cujas classes minoritárias sejam substituidas pela classe marjoritária. Observando a **Figura 3**, da matriz de confusão do SVM, podemos perceber que o modelo pontua com acertos nas classes minoritárias, obtendo assim uma acurácia superior a um modelo totalmente marjoritário.

Além da geração de dados sintéticos, outras abordagens para mitigar o viés no Datareq incluem a redução de classes majoritárias (undersampling) e a utilização de pesos para aumentar a relevância de classes minoritárias. A técnica de undersampling consiste em reduzir o número de instâncias das classes majoritárias, buscando um equilíbrio maior entre as classes. No entanto, essa abordagem pode levar à perda de informações valiosas presentes nos dados descartados. A utilização de pesos para classes desbalanceadas, por sua vez, atribui pesos maiores às classes minoritárias durante o treinamento do modelo. Dessa forma, o modelo é penalizado com mais intensidade por erros de classificação nessas classes, o que as torna mais relevantes no processo de aprendizado. Neste estudo, exploramos a técnica de pesos para classes desbalanceadas com o objetivo de aumentar a importância das categorias menos representadas no Datareq. Um modelo SVM com kernel linear foi treinado com o uso de pesos para classes desbalanceadas, resultando em um valor de acurácia de 50% para a tarefa de classificação unitária de categorias.

5. Conclusão

Este trabalho apresentou o Datareq, um conjunto de dados contendo 1502 requisitos em português extraídos de documentos públicos disponíveis na internet. Cada requisito foi

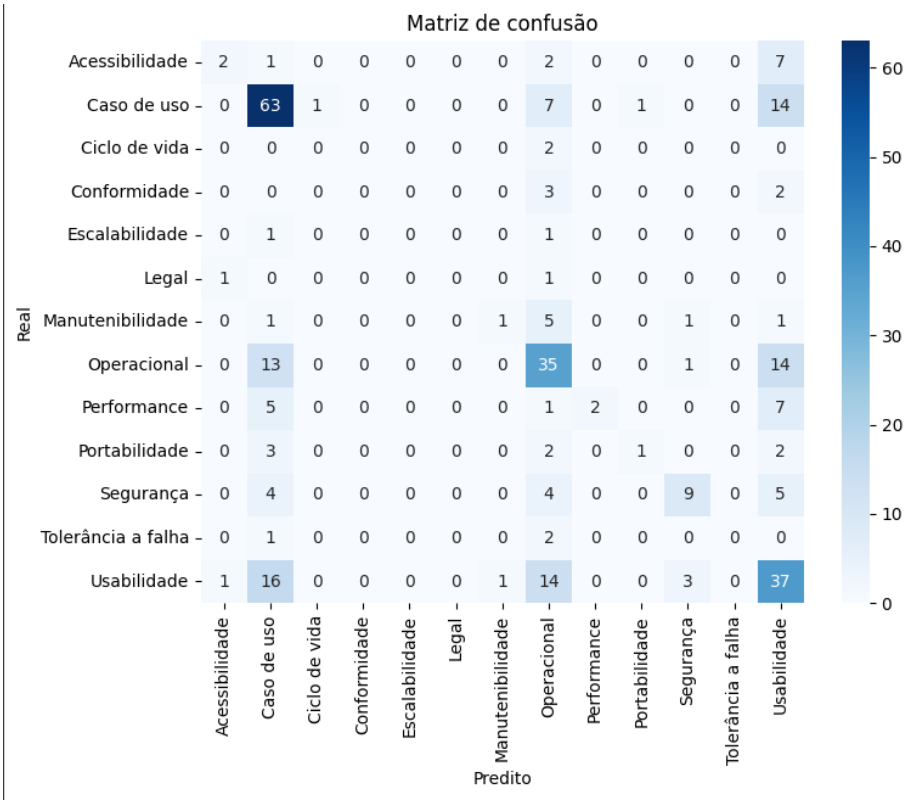

Figure 3. Matriz de confusão do SVM

rotulado manualmente quanto ao seu tipo (funcional ou não funcional), categoria (em um conjunto de 13 categorias) e linguagem (sistema). O Datareq se destina ao treinamento de modelos de aprendizado supervisionado e processamento de linguagem natural (PLN). Os testes de classificação de funcionalidade e categorias demonstraram a utilidade do Datareq, embora tenham revelado a presença de um viés significativo na distribuição dos dados.

Como trabalhos futuros, propomos o aprimoramento do Datareq em duas frentes principais: linguagem e variância. Inicialmente, o objetivo era avaliar requisitos com linguagem técnica (sistema) e linguagem usual (usuário). No entanto, a análise dos editais resultou na extração exclusiva de requisitos com linguagem de sistema. Assim, uma próxima etapa consiste na extração de requisitos de fontes menos formais, como manuais de usuário ou documentação de projetos, para obter dados com linguagem de usuário e, assim, diversificar o conjunto de dados quanto a esse aspecto.

Em relação à variância, o Datareq apresenta uma distribuição desbalanceada entre as categorias e uma baixa representatividade de requisitos não funcionais. Para mitigar esse problema, pretendemos explorar a geração de dados sintéticos para complementar tanto os requisitos não funcionais quanto as categorias minoritárias, buscando um conjunto de dados mais equilibrado e representativo.

References

A. Ferrari, G. O. S. and Gnesi, S. (2017). Pure: A dataset of public requirements documents. IEEE 25th International Requirements Engineering Conference (RE).

Abdur Rahman, Abu Nayem, S. S. (2023). Non-functional requirements classification using machine learning algorithms. International Journal of Intelligent Systems and Applications(IJISA).

de Araújo, A. F. and Marcacini, R. M. (2021). Re-bert: automatic extraction of software requirements from app reviews using bert language model. 36th Annual ACM Symposium on Applied Computing (SAC '21).

Faceli, K; Lorena, A. C. G. J. A. T. A. C. A. C. (2021). Inteligencia artificial. In *Uma Abordagem de Aprendizado de Máquina*. gen.

G. Y. Quba, H. Al Qaisi, A. A. and AlZu'bi, S. (2021). Software requirements classification using machine learning algorithm's. International Conference on Information Technology (ICIT).

Lu, M. and Liang, P. (2017). Automatic classification of non-functional requirements from augmented app user reviews. International Conference on Evaluation and Assessment in Software Engineering (EASE '17).

Senger, T. (2022). A unified open-and closed-source software requirements dataset. Universität Stuttgart.

Shukla, V. (2024). Software requirements dataset.

SOMMERVILLE, I. (2011). Engenharia de software. In Hirama, K., editor, *Engenharia de software*. Pearson.

Souvik (2020). Software requirements dataset.

T. Hey, J. Keim, A. K. and Tichy, W. F. (2020). Norbert: Transfer learning for requirements classification,. 2020 IEEE 28th International Requirements Engineering Conference (RE).

Yucalar, F. (2023). Developing an advanced software requirements classification model using bert: An empirical evaluation study on newly generated turkish data. MDPI Applied Sciences.