Busca e Recuperação de *Workflows* em Repositórios por meio de *Transformers* e Modelagem de Tópicos

Lyncoln S. Oliveira[1], Annie Amorim[1], Marcos Lage[1], Aline Paes[1], Daniel de Oliveira[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)

{lyncolnsousa,annieamorim}@id.uff.br, {mlage,alinepaes,danielcmo}@ic.uff.br

Abstract. *Various repositories provide pre-modeled workflows for reuse and adaptation, given the inherent complexity of workflow modeling. Although these repositories offer labeling mechanisms, such labels are not always filled in, and when they are, their values can limit the search. An alternative way to perform searches in these repositories is to use natural language descriptions of workflows rather than being restricted to label-based searches or structural comparisons of workflows, which may be unfeasible. This paper presents the* `Athena++` *approach, which uses natural language processing techniques to search for workflows in repositories, specifically using Transformers and Topic Modeling. The* `Athena++` *was evaluated with a set of workflows obtained from the Galaxy repository, and the results were promising.*

Resumo. *Diversos repositórios disponibilizam workflows previamente modelados para reúso e adaptação, dada a complexidade inerente à modelagem de um workflow. Embora esses repositórios ofereçam mecanismos de rotulação, nem sempre tais rótulos são preenchidos e, quando o são, os valores informados acabam limitando a busca. Um modo alternativo de realizar a busca nesses repositórios é utilizar as descrições em linguagem natural dos workflows, em vez de se limitar à busca por rótulos ou à comparação estrutural dos workflows. Este artigo apresenta a abordagem* `Athena++`*, que utiliza técnicas de processamento de linguagem natural para realizar a busca por workflows em repositórios, em especial o uso de Transformers e Modelagem de Tópicos. A* `Athena++` *foi avaliada com um conjunto de workflows obtidos no repositório do Galaxy, e os resultados se mostraram promissores.*

1. Introdução

Nas últimas décadas, o uso de *workflows* como uma abstração para modelar experimentos científicos baseados em simulações tornou-se um padrão [de Oliveira et al. 2019]. A modelagem de *workflows* em um sistema de *workflows* ou via *scripts* é um processo iterativo e complexo de aprendizado, onde o reúso é uma questão crucial. Enfrentar esse desafio é fundamental, pois permite aos pesquisadores otimizar seus esforços e aprimorar a qualidade e a eficácia de seus experimentos. Portanto, os usuários podem se beneficiar de *workflows* previamente modelados, em vez de começarem a especificação do zero [Dias et al. 2024]. Existem diversos repositórios públicos de *workflows*, como o *myExperiment* [Goble et al. 2010] e o *Galaxy ToolShed* [Blankenberg et al. 2014]. Apesar de cada um deles possuir funcionalidades específicas, ambos oferecem mecanismos de rotulação para os *workflows* armazenados, além de permitirem que o usuário "dono" do *workflow* associe um conjunto de metadados a esses *workflows*. Tais rótulos e metadados são utilizados por terceiros para realizar recuperar os *workflows* para reúso. Entretanto, realizar buscas exclusivamente por

meio de rótulos pode apresentar limitações. Em primeiro lugar, porque os rótulos podem não refletir o que o *workflow* executa como um todo. Além disso, muitos usuários não rotulam seus *workflows* ou adicionam metadados. No repositório *Galaxy ToolShed*, de 1.014 *workflows* registrados, apenas 187 possuem pelo menos um rótulo. Diversos trabalhos na literatura têm focado em buscar *workflows* em repositórios para facilitar o reúso [Silva et al. 2011, Zhou et al. 2018, Starlinger et al. 2016]. A maioria desses trabalhos concentra-se em comparar a estrutura dos *workflows* a fim de encontrar similaridades entre eles. Entretanto, essa é uma tarefa complexa, pois cada *workflow* pode ser especificado em sistemas diferentes que seguem linguagens de especificação distintas. Isso demanda que os *workflows* sejam convertidos para uma representação canônica, o que pode não ser trivial devido à heterogeneidade das representações.

Uma alternativa à busca por meio da estrutura do *workflow* é utilizar as descrições textuais dos *workflows*. A busca por descrições detalhadas fornece uma compreensão mais precisa do propósito e funcionalidade dos *workflows*. Diferentemente dos rótulos, grande parte dos *workflows* nos repositórios possui uma descrição em linguagem natural associada à especificação do *workflow*, e essa descrição pode ser uma fonte rica de informação para buscas. Buscar *workflows* por meio de textos em linguagem natural não é uma abordagem nova [Costa et al. 2012, Gu et al. 2023], mas ganhou maior relevância nos últimos anos, principalmente devido a novas técnicas de Processamento de Linguagem Natural (PLN), como o uso de *Transformers* e Modelagem de Tópicos. O artigo apresenta a `Athena++`, uma abordagem para buscas de *workflows* em repositórios por meio de PLN. A `Athena++` estende a abordagem `Athena` [Costa et al. 2012] ao incorporar novas técnicas de PLN, como o uso de *Transformers* e Modelagem de Tópicos. A `Athena++` foi avaliada utilizando *workflows* reais obtidos no repositório do sistema de *workflow* Galaxy, e os resultados mostraram-se promissores, tanto em relação à qualidade dos resultados quanto ao desempenho.

Este artigo está organizado em quatro seções, além da Introdução. A Seção 2 apresenta uma breve descrição sobre modelos de linguagem e modelagem de tópicos, além de apresentar os trabalhos relacionados. A Seção 3 apresenta a abordagem `Athena++`. A Seção 4 discute a avaliação experimental e, por fim, a Seção 5 conclui o artigo.

2. Referencial Teórico e Trabalhos Relacionados

Nesta seção, discutimos dois conceitos fundamentais para o presente artigo: os Modelos de Linguagem e a Modelagem de Tópicos (Subseção 2.1) Além disso, na Subseção 2.2 são brevemente discutidos os trabalhos relacionados.

2.1. Modelos de Linguagem e Modelagem de Tópicos

Os Modelos de Linguagem (ML) são sistemas de Inteligência Artificial (IA) treinados com bilhões de termos. Esse processo de treinamento, que pode envolver múltiplas etapas e diferentes níveis de intervenção humana, permite que os modelos aprendam como os termos interagem entre si em diversos contextos linguísticos. Consequentemente, esses modelos são capazes de identificar e replicar padrões de linguagem, aplicando-os em uma variedade de tarefas de PLN [Thirunavukarasu et al. 2023]. Dentro deste espectro, os *Transformers* [Vaswani et al. 2017] surgem como uma arquitetura de redes neurais projetada para processar dados sequenciais, como textos. Esta arquitetura se destaca por sua capacidade de lidar simultaneamente com todas as partes de um texto de entrada, permitindo uma compreensão mais profunda e contextual do mesmo. O BERT (*Bidirectional Encoder Representations from Transformers*), baseado nesta arquitetura, foi desenvolvido especificamente

para o treinamento de modelos de linguagem em grandes volumes de texto de forma auto-supervisionada [Souza et al. 2020]. Isso envolve a análise de textos não rotulados, permitindo que o modelo aprenda independentemente a estrutura e o significado do texto, potencializando sua eficácia em diversas aplicações de PLN.

A Modelagem de Tópicos (MT) é uma técnica de PLN que visa extrair, resumir e categorizar informações de conjuntos de textos [Blei 2012]. Ao identificar padrões semânticos nos textos, ela permite a categorização e síntese de informações, tornando mais acessível a interpretação de dados. Dentre os métodos de MT, o BERTopic[1] destaca-se como uma abordagem para identificar automaticamente tópicos densos em coleções de documentos [Grootendorst 2022]. O BERTopic se beneficia dos *embeddings*, que são representações numéricas de termos [Reimers and Gurevych 2019], e também faz uso da valoração *Term Frequency-Inverse Document Frequency* (TF-IDF) para criar tópicos interpretáveis. O algoritmo do BERTopic opera em três etapas: (i) cada documento é convertido em sua representação de *embedding* usando um modelo linguístico pré-treinado, (ii) antes de agrupar os *embeddings*, sua dimensionalidade é reduzida para facilitar o processamento, e (iii) a partir dos tópicos dos documentos, as representações dos tópicos são extraídas usando uma variação personalizada do método TF-IDF.

2.2. Trabalhos Relacionados

O tópico de recuperação de *workflows* em repositórios não é recente e tem sido alvo de pesquisas ao longo da última década. As abordagens existentes podem ser classificadas em três tipos: (i) baseadas na estrutura do *workflow*, (ii) baseadas em metadados e (iii) híbridas. A primeira categoria engloba trabalhos que buscam *workflows* com estruturas similares. [Silva et al. 2011] propõem um arcabouço que analisa a estrutura do *workflow* e, por meio de heurísticas, define um índice de similaridade entre os *workflows*, facilitando assim a busca. De maneira similar, [Starlinger et al. 2016] e [Zhou et al. 2018] propõem arquiteturas que extraem uma representação canônica do *workflow* para cálculo das similaridades entre *workflows*. Apesar de tais abordagens se mostrarem eficientes na busca por *workflows* com estruturas similares, ao lidar com *workflows* especificados em múltiplas linguagens, elas requerem um *parser* específico, o que pode ser um impeditivo. Além disso, essas abordagens não consideram a semântica por trás do *workflow* e suas atividades, *e.g.*, não permitem que um usuário recupere *workflows* que realizam "*alinhamento múltiplo de sequências*" a menos que o nome da atividade seja o mesmo ou que se saiba *a priori* quais programas implementam o alinhamento.

A segunda categoria de abordagens concentra-se na recuperação de *workflows* por meio de metadados, sejam eles estruturados ou não, como descrições do *workflow* em linguagem natural. O arcabouço `Athena` [Costa et al. 2012] aplica técnicas de mineração de texto para descobrir *workflows*, focando na extração de descrições em linguagem natural a partir de diferentes repositórios. O `Athena` baseava-se exclusivamente na geração de *Bag-of-Words* (BoW) para agrupamento com o algoritmo *K-Means*, o que apresentava limitações tanto em escalabilidade quanto na qualidade dos resultados. Na terceira categoria, a abordagem SWORTS [Gu et al. 2023] combina semântica textual e estrutural utilizando um modelo hierárquico para a recuperação de *workflows*. No entanto, a SWORTS utiliza apenas uma técnica de PLN e baseia-se exclusivamente na descrição para identificar *workflows* similares, o que pode restringir a eficácia das buscas. A abordagem `Athena++`, proposta neste

[1] https://maartengr.github.io/BERTopic/index.html

artigo, tem como objetivo estender a `Athena` para aplicar diferentes técnicas de PLN para a recuperação de *workflows*, sem depender da análise de sua estrutura, o que permite sua aplicação a *workflows* modelados em diversos formatos e padrões. Tal característica torna a `Athena++` mais flexível, permitindo uma recuperação mais abrangente. Além disso, a combinação de diferentes técnicas de PLN no `Athena++` busca superar limitações de abordagens anteriores, proporcionando uma análise semântica mais profunda, o que pode aumentar a eficácia na identificação de *workflows* similares, mesmo quando as descrições textuais são limitadas ou variam em terminologia.

3. Abordagem Proposta: `Athena++`

A arquitetura da `Athena++` é apresentada na Figura 1 e é composta por três camadas: (i) Repositórios de *Workflows*, (ii) Camada de Processamento e (iii) Camada de Dados. A Camada de Repositórios contém todos os repositórios de *workflows* que podem ser acessados pela `Athena++`. O único requisito desses repositórios é que possuam uma API de acesso ou que seus dados possam ser extraídos de páginas *web* estáticas. Na Camada de Processamento, o componente *Crawler* acessa cada um dos repositórios por meio de APIs ou utilizando ferramentas de *web scraping* para extrair a especificação dos *workflows* (*i.e.*, arquivos JSON) e os metadados disponíveis, tanto os estruturados quanto os em linguagem natural. Os dados obtidos são armazenados na Camada de Dados, que contém tanto o repositório de metadados quanto as especificações dos *workflows*.

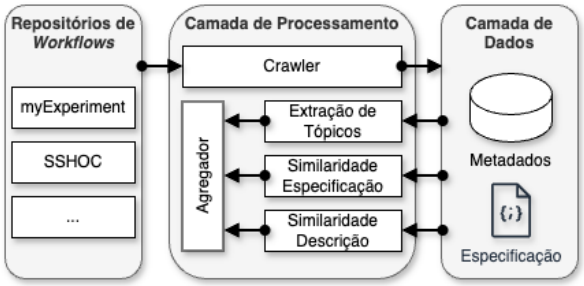

Figura 1. Arquitetura da `Athena++`

Uma vez que os *workflows* já foram obtidos, a similaridade pode ser então calculada. Para identificar os *top* n *workflows* similares, foram definidos três componentes na Camada de Processamento: (i) Similaridade de Descrição, (ii) Similaridade de Especificação e (iii) Extração de Tópicos. O componente *Similaridade de Descrição* considera apenas a descrição em linguagem natural informada pelo usuário. Nele, o arquivo JSON de um *workflow* é transformado em texto, e informações relevantes, como nomes, descrições e rótulos, são extraídas. Essas informações são então combinadas em um único documento representativo para cada *workflow*. Utilizando o BERT, o documento associado a cada *workflow* é convertido em *embeddings*, que são vetores que capturam o significado semântico do texto. Para cada *workflow*, calcula-se a média dos *embeddings* de todos os *tokens* presentes no texto, criando uma representação densa e compacta. Com as representações de todos os *workflows*, é construída uma matriz de similaridade utilizando a similaridade de cosseno. Essa matriz permite comparar cada *workflow* com todos os outros, identificando aqueles com maior similaridade. Para cada *workflow*, são identificados os *top* n *workflows* mais similares.

Em paralelo, o componente *Similaridade de Especificação* utiliza todo o arquivo JSON como documento representativo. Sua execução é similar ao do componente *Similaridade de Descrição*, entretanto sem o pré-processamento realizado para extração de nomes e

descrições. Apesar do componente *Similaridade de Especificação* não considerar diferenças nas linguagens de especificação, essa comparação se mostra útil, pois *workflows* similares podem invocar os mesmos programas, e esse nível de similaridade será identificado.

O componente *Extração de Tópicos* segue o mesmo processo de extração dos metadados do componente *Similaridade de Descrição*; entretanto, utiliza o BERTopic para identificar os principais *tokens* dos textos. A extração de tópicos dos metadados do *workflow* é um dos diferenciais da `Athena++` em relação aos trabalhos relacionados. Para cada *workflow*, aplica-se o BERTopic, extraindo-se então os 10 termos principais do primeiro tópico identificado. Caso haja mais de um tópico identificado inicialmente, o modelo combina tópicos semelhantes ou menos representativos até que apenas um tópico principal permaneça. Se o texto do *workflow* possuir menos ou apenas 10 termos, ele é considerado o próprio tópico. Os termos dos tópicos são convertidos em um único texto, que é então transformado em *embeddings* utilizando o modelo BERT. Esses *embeddings* representam a estrutura semântica dos tópicos identificados. Finalmente, a partir das similaridades calculadas por cada um dos componentes mencionados anteriormente, o *Agregador* define quais são os *top* n *workflows* similares que serão recuperados pelo usuários. O código-fonte da `Athena++` será disponibilizado no repositório do GitHub `https://github.com/UFFeScience/athena-`.

4. Avaliação Experimental da `Athena++`

Esta seção apresenta a avaliação experimental da `Athena++` de forma a analisar a utilidade das similaridades calculadas e o desempenho da `Athena++`.

4.1. Configuração dos Experimentos e do Ambiente de Execução

Para os experimentos, foram coletados *workflows* do repositório *Galaxy ToolShed*, que possui *workflows* da bioinformática. O conjunto inclui 1.014 *workflows*, disponibilizados no formato JSON. Um subconjunto com 187 *workflows* possui metadados adicionais, como nome, descrição e rótulo, que descrevem as funcionalidades implementadas no *workflow* e ajudam na categorização e identificação dentro do repositório. Esses atributos podem se aplicar ao *workflow* como um todo ou estar relacionados a uma atividade dentro do *workflow*, oferecendo detalhes adicionais sobre a função da mesma.

Para o cálculo da similaridade e recuperação dos *workflows*, foram usados dois modelos de linguagem baseados em *Transformers*, com o objetivo de explorar tanto contextos gerais quanto específicos da bioinformática. O primeiro modelo utilizado foi o BERT, que é conhecido por sua capacidade de entender o contexto de palavras em textos em inglês de maneira geral, sendo projetado para uma abordagem ampla. Em contrapartida, para uma análise mais focada no jargão científico da bioinformática, o modelo SciBERT[2] foi também utilizado. Este modelo é uma variação do BERT, treinado especificamente em um vasto *corpus* de textos científicos, o que o torna adequado para compreender e processar terminologias e conceitos especializados da área. A tokenização do texto teve seu valor máximo definido como 512 *tokens* para compatibilidade com o SciBERT e o BERT. Isso inclui a aplicação de *padding* para textos menores e truncamento para os que excedem o limite, garantindo uniformidade nos dados de entrada. Após a tokenização, o modelo de linguagem processa os *tokens* sem realizar atualizações de peso, de modo a obter os *embeddings* correspondentes a cada *token* do texto.

[2]https://huggingface.co/allenai/scibert_scivocab_uncased

Na modelagem de tópicos, os mesmos modelos de linguagem para *embeddings* foram utilizados, integrados ao BERTopic. Dado o tamanho reduzido dos textos que descrevem os *workflows* coletados, o algoritmo HDBSCAN foi ajustado para um *cluster* de tamanho mínimo dois. Além disso, a redução de dimensionalidade foi realizada por meio do UMAP, definido para duas componentes principais, a fim de construir tópicos mesmo com a limitada quantidade de dados disponíveis. O ambiente computacional utilizado nos experimentos foi uma máquina com processador i5-8400, 19 GB RAM, GPU RTX 3070 e sistema operacional *Windows Subsystem for Linux* com imagem do Ubuntu 20.04.6 LTS.

4.2. Resultados

Avaliamos o desempenho da `Athena++` utilizando tanto o BERT quanto o SciBERT. Para verificar a qualidade dos resultados da `Athena++`, utilizamos como oráculo os 187 *workflows* que possuem rótulos. O objetivo é comparar os resultados retornados pela `Athena++` com os rótulos definidos pelos usuários, *i.e.*, verificar se os rótulos dos *workflows* retornados como similares estão no conjunto de rótulos do *workflow* cuja descrição foi fornecida como entrada. Para cada *workflow* (*i.e.*, sua especificação e descrição textual), buscamos os 3 a 10 *workflows* mais similares. Para cada um dos modelos de linguagem, realizamos 10 execuções e calculamos a média dos tempos de execução e do consumo de CPU/GPU. A Tabela 1 apresenta a quantidade de *workflows* recuperados pela `Athena++` cujos rótulos se assemelham aos do *workflow* cuja descrição foi fornecida como entrada. Observamos que, no pior caso (*top* 3 *workflows* mais similares usando o BERT), a `Athena++` foi capaz de recuperar 60% dos *workflows* similares de acordo com o oráculo. Ao combinarmos os *workflows* similares encontrados tanto com o BERT quanto com o SciBERT, obtivemos 119 acertos no pior caso (*top* 3 *workflows* mais similares) e 132 acertos no melhor caso (*top* 10 *workflows* mais similares), resultando em uma taxa de acerto de 71%.

Tabela 1. Quantidade total de *workflows* similares recuperados para os modelos de linguagem BERT e SciBERT.

Modelo de Linguagem	Top	Descrição	Tópicos	Combinação Métodos
BERT	3	106	47	113
BERT	4	106	58	113
BERT	5	106	67	116
BERT	6	106	62	117
BERT	7	106	63	115
BERT	8	106	66	114
BERT	9	106	71	118
BERT	10	107	64	117
SciBERT	3	100	72	112
SciBERT	4	100	73	114
SciBERT	5	102	73	117
SciBERT	6	102	83	114
SciBERT	7	102	82	117
SciBERT	8	103	77	115
SciBERT	9	104	87	118
SciBERT	10	105	84	116

Além dos resultados qualitativos, analisamos também o tempo de execução para retornar de 3 a 10 *workflows* similares para cada um dos 187 *workflows* do oráculo. A Figura 2 apresenta os tempos de execução em segundos para cada cálculo de similaridade usando o BERT e o SciBERT. Observamos que a similaridade baseada na extração de tópicos é a etapa mais onerosa do processo, independentemente do modelo de linguagem escolhido, dominando o tempo de execução total. Apesar de ser uma etapa custosa, a extração de tópicos possibilitou a recuperação de um conjunto de *workflows* que, com base na descrição e na estrutura, não seriam definidos como similares.

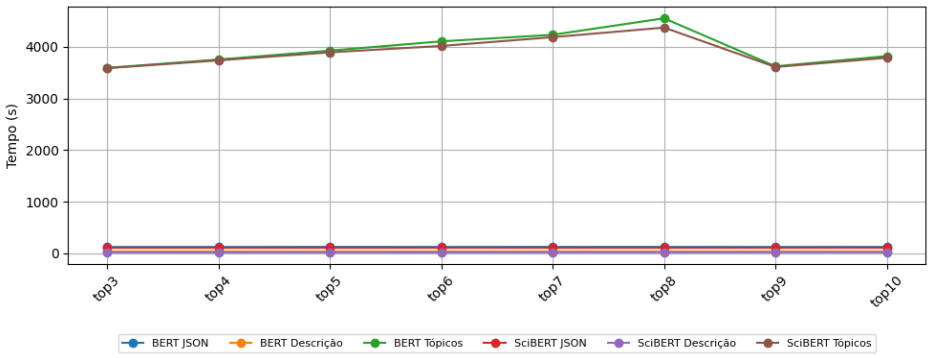

Figura 2. Tempos de execução da `Athena++` por configuração.

As Figuras 3(a) e 3(b) apresentam o consumo de memória tanto para CPU quanto da GPU para execução da `Athena++`. As Figuras 3(c) e 3(d) apresentam o percentual de CPU e GPU consumidos, respectivamente. Tais resultados evidenciam o bom desempenho da `Athena++`, e que o uso da GPU permite uma aceleração significativa nos modelos de linguagem, enquanto os modelos baseados em tópicos, que utilizam apenas a CPU, apresentam um maior custo de tempo de processamento.

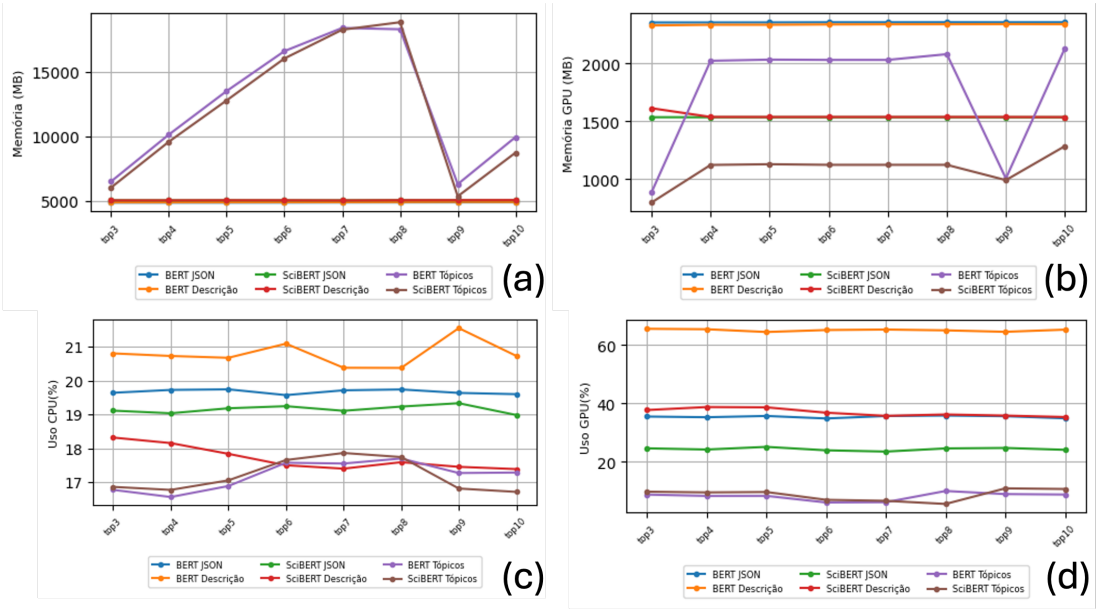

Figura 3. Consumo de memória e uso de CPU/GPU na `Athena++`.

5. Conclusão

Modelar um *workflow* pode não ser uma tarefa trivial. A definição do *dataflow* e a integração de múltiplos *softwares* e bibliotecas dependem da expertise do usuário. O reúso de *workflows* já definidos pode ajudar nessa tarefa complexa. Muitos repositórios disponibilizam um conjunto de *workflows* submetidos pela comunidade científica. Entretanto, recuperar um *workflow* nesses repositórios pode não ser uma tarefa simples. Para facilitar a recuperação de *workflows* em repositórios distribuídos e, assim, fomentar o reúso, apresentamos neste artigo a abordagem `Athena++`, que visa recuperar *workflows* em diferentes repositórios, como o *myExperiment* e o *Galaxy ToolShed*, analisando descrições em linguagem natural e a estrutura do *workflow* para encontrar similaridades.

Embora tenhamos utilizado uma quantidade moderada de *workflows* na avaliação experimental (187 *workflows* com descrições), os resultados evidenciaram que os modelos de linguagem e de tópicos são uma solução promissora para calcular a similaridade entre *workflows*. Embora a `Athena++` represente um avanço, sua robustez é suscetível a alguns aspectos. O primeiro é a quantidade de descrições de *workflows* disponíveis. O segundo aspecto é relativo ao estilo de escrita do usuário. Cada usuário tem um estilo de escrita particular e utiliza um conjunto específico de palavras em seus textos. Como trabalho futuro, planejamos gerar *clusters* que possam ser usados para modelar representações de abstrações de alto nível para *workflows*, como Linhas de Experimentos [Dias et al. 2024]. Ademais, consideramos a integração de estruturas semânticas mais robustas para enriquecer a abordagem.

Referências

Blankenberg, D. et al. (2014). Dissemination of scientific software with galaxy toolshed. *Genome Biology*, 15(2):403.

Blei, D. M. (2012). Probabilistic topic models. *Commun. of the ACM*, 55(4):77–84.

Costa, F. et al. (2012). Athena: text mining based discovery of scientific workflows in disperse repositories. In *RED 2010, Paris, France*, pages 104–121. Springer.

de Oliveira, D., Liu, J., and Pacitti, E. (2019). *Data-Intensive Workflow Management: For Clouds and Data-Intensive and Scalable Computing Environments*. Morgan & Claypool.

Dias, L. G. et al. (2024). Maestro: a lightweight ontology-based framework for composing and analyzing script-based scientific experiments. *Knowledge and Information Systems*.

Goble, C. A. et al. (2010). myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.*, 38:677–682.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *CoRR*, abs/2203.05794.

Gu, Y., Cao, J., Qian, S., and Guan, W. (2023). Sworts: a scientific workflow retrieval approach by learning textual and structural semantics. *IEEE Trans. on Services Computing*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. pages 3980–3990.

Silva, V. et al. (2011). Similarity-based workflow clustering. *Journal of Computational Interdisciplinary Sciences*, 2(1):23–35.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In *Intelligent Systems*, pages 403–417, Cham. Springer International.

Starlinger, J. et al. (2016). Effective and efficient similarity search in scientific workflow repositories. *Future Generation Computer Systems*, 56:584–594.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. (2023). Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Zhou, Z., Cheng, Z., Zhang, L.-J., Gaaloul, W., and Ning, K. (2018). Scientific workflow clustering and recommendation leveraging layer hierarchical analysis. *IEEE Transactions on Services Computing*, 11(1):169–183.