Visualização Interativa da Evolução de Grafos de Conhecimento

Eduardo Moreira Freitas de Souza[1] **, Anderson Rossanez**[1] **,**
Julio Cesar dos Reis[1] **, Ricardo da Silva Torres**[2,3]

[1]Institute of Computing, University of Campinas (Unicamp), Campinas, Brazil

eduardomoreirafds@gmail.com,{anderson.rossanez,jreis}@ic.unicamp.br

[2]NTNU – Norwegian University of Science and Technology, Ålesund, Norway

ricardo.torres@ntnu.no

[3]Wageningen University & Research, Wageningen, The Netherlands

ricardo.dasilvatorres@wur.nl

Resumo. *Grafos de conhecimento são artefatos essenciais na representação estruturada de dados em diversos domínios de aplicação. Grafos de conhecimento temporais expressam um conjunto de tais grafos no tempo. Uma necessidade relevante reside na análise e tomada de decisões a partir dessas estruturas. Estudamos a concepção de uma ferramenta para a visualização interativa da evolução de grafos de conhecimento. Implementamos TKGEvolViewer, uma ferramenta que possibilita a exploração gráfica de grafos de conhecimento temporais a partir de valores de métricas codificadas em suas estruturas. Nossos resultados permitem a condução de análises visuais sobre os grafos através de um modal gráfico, filtrando informações obtidas de análises predefinidas.*

Abstract. *Knowledge graphs are essential artifacts employed to represent structured data in several application domains. Temporal knowledge graphs express a set of knowledge graphs that change over time. A relevant need for such structures is how one makes use of them to analyze knowledge and make decisions. This investigation studies the conception and development of a software tool, for interactively visualizing the evolution of knowledge graphs. We implemented the TKGEvolViewer tool, which enables graphically exploring temporal knowledge graphs, by means of measurements coded within their structures. Our results enable users to conduct visual analyses over temporal knowledge graphs, filtering available information through a graphic modal.*

1. Introdução

Representação visual de dados pode ajudar a sintetizar aspectos relevantes de uma coleção. Por exemplo, um histograma apresenta os componentes mais frequentes, enquanto uma série temporal codifica a variação de alguma variável em relação ao tempo. Há um grande desafio no *design* de meios para a representação de maneira que satisfaça a necessidade do usuário aproveitando-se ao máximo das propriedades dos dados.

Em cenários mais simples, como descobrir a tendência do custo de um produto, ou a variação do preço em relação ao tempo, representações existentes são em geral suficientes. Por outro lado, um cenário de análise de dados a partir da visualização de objetos

e suas relações (por exemplo, contatos de um usuário em uma rede social) ao longo do tempo é mais desafiador. Modelar adequadamente os objetivos da análise, bem como escolher a melhor representação dos dados a fim de atender os requisitos de análise são desafios típicos.

Nesse exemplo, uma das possibilidades em estruturar os dados de um determinado momento é através de um *grafo*. Usuários podem ser representados como vértices, e as suas conexões como arestas. Ademais, é possível que existam várias versões desse grafo, representando essas conexões em diferentes momentos no tempo. Nesse cenário, não existe apenas a questão de como representar os grafos em sua unidade de tempo, mas igualmente o caso de representar as suas mudanças a cada iteração, aumentando a complexidade do problema da visualização.

De maneira similar, conhecimento pode ser representado através de grafos. Os grafos de conhecimento (*Knowledge Graphs* [Hogan et al. 2021] – KGs) utilizam um modelo de dados baseado em grafos para capturar conhecimento de diversas fontes em larga escala. Análises baseadas em KGs são utilizadas em diversos tipos de sistemas, tais quais, busca por informação [Arnaout and Elbassuoni 2018], sistemas de reposta à questões [Singh et al. 2020], extração de informação de notícias [Rospocher et al. 2016], assistentes [Kim 2017], dentre diversos outros. Conforme o conhecimento evolui ao longo do tempo [Chen et al. 2008], é natural que possam existir diversas versões de um KG, as quais representam o estado do conhecimento naquele instante específico do tempo.

Este estudo propõe e desenvolve meios de representar e explorar graficamente um grupo específico de grafos dinâmicos, conhecidos como *Temporal Knowledge Graphs* (TKGs) [Rossanez et al. 2020a]. Enquanto KGs são compostos por triplas *RDF* (*Resource Description Framework*), na forma de sujeito—predicado—objeto (*i.e.* conceitos são ligados através relações, como em Fernando—visitou—Espanha), os TKGs são compostos por mais de uma iteração de um *KG*. Dessa forma, um *TKG* contém representações das triplas de um determinado domínio em diversos momentos no tempo.

Há diversas aplicações da visualização de grafos que evoluem no tempo. Por exemplo, [Beck et al. 2017] estudaram o mapeamento de interações entre proteínas na biologia, transações econômicas no mundo financeiro ou até mesmo performance de jogadores nos esportes. Rossanez *et al.* [Rossanez et al. 2020a] aplicaram a visualização de KGs em na análise da literatura biomédica sobre doenças degenerativas. Nenhuma destas formulações existentes explora, em específico, a visualização de TKGs. Tampouco exploram na visualização, métricas que codificam propriedades dos TKGs.

Neste estudo, visamos aprimorar a compreensão sobre a análise de TKGs através do desenvolvimento de uma ferramenta de software original. Nossa proposta explora interação visual com os TKGs através de um fluxo de uso que guia os usuários. Nossa proposta permite análises através da exploração de métricas referentes à própria estrutura gráfica dos vértices.

Nossa principal contribuição é o *TKGEvolViewer*, uma ferramenta que permite a visualização interativa de *TKGs*, com seus *KGs* separados espacialmente em ordem cronológica — em formato de linha do tempo, representando seus nós (sujeitos e objetos) com círculos e suas arestas (predicados) com segmentos de retas. As setas são devidamente coloridas de acordo com qual iteração ele representa. A ferramenta permite a

exibição das métricas extraídas e possibilidade de livre exploração do grafo, visualizando a estrutura dos conceitos através de *zoom-in*, *zoom-out* e translação na representação, além do etiquetamento e realce dos vértices, bem como de suas arestas.

O restante deste artigo está organizado da seguinte maneira: A Seção 2 discute estudos sobre o uso de KGs e técnicas de visualização de Grafos Dinâmicos. A Seção 3 detalha o fluxos de uso e uma visão arquitetural da ferramenta. A Seção 4 apresenta um estudo de caso, detalhando a execução da ferramenta para um TKG extraído de dados reais. A Seção 5 discute nossos resultados, enquanto a Seção 6 apresenta as conclusões.

2. Visualização de Grafos de Conhecimento

Conduzimos uma síntese de trabalhos da literatura considerando *usos de KGs*, *análise de grafos dinâmicos* e *técnicas de visualização de grafos dinâmicos*. Rossanez *et al.* [Rossanez and Dos Reis 2019] propuseram uma técnica semiautomática de geração de *KGs* a partir da literatura científica de doenças degenerativas, como Alzheimer. Eles abordaram o processo de criação de um *KG*. A proposta consistiu em quatro módulos: um preprocessador, responsável por simplificar as orações do texto original, resolver abreviações e pronomes. O texto pré-processado é enviado ao extrator de triplas. Esse extrai candidatos a sujeitos, predicados e objetos, além de ser responsável por reconhecer orações na voz passiva e identificar agentes ou pacientes possivelmente fora da oração, de forma que a tripla contenha o elemento original, e não uma referência a este. A partir do texto preprocessado, é gerada uma *parse tree* de cada oração, identificando os elementos desta com uma etiqueta de uso semântico e os ligando a um elemento de uma Ontologia. Por último, a ferramenta gera um arquivo *turtle* que contém um recurso para cada constituinte da tripla — bem como a sua ligação com a respectiva Ontologia — e o relacionamento entre cada elemento desta: o resultado é a descrição de um *KG*.

Pomp *et al.* [Pomp et al. 2020] propuseram um uso diferente para KGs. Sua abordagem tem como objetivo identificar de forma automática o tipo do dado inserido em um sistema para se manter um bom nível de organização na manutenção de um grande número de dados armazenados. A proposta faz uso de um *KG* para que se armazene a escolha mais provável entre texto, identificador, *bag of words* e valor numérico. O tipo de dado é um nó ligado a cada classificação por uma aresta ponderada, que indica o resultado do modelo de treinamento para a base de dados proposta.

Rossanez et al. [Rossanez et al. 2020a] propuseram um estudo de métricas de centralidades extraídas dos *TKGs*. As métricas centralidade exploradas são: *degree centrality*, *eigenvector centrality* e *betweenness centrality*. Através delas, é possível representar a mudança de conhecimento acerca de diferentes iterações de corpos textuais de um mesmo tema ou contexto, porém em tempos diferentes.

Tosi e Dos Reis [Tosi and dos Reis 2022] propuseram um estudo acerca da evolução de um campo científico a nível de conceito, identificando subáreas e as suas relações, assim como os conceitos pertencentes a elas, bem como as relações entre si. Apresentaram uma ferramenta de software que auxilia o processo, *SciKGraph*. Esta ferramenta é responsável pela construção do *KG*, que conecta conceitos que se apresentam próximos no texto original com arestas ponderadas, que têm seu peso ajustado de acordo com o número de ocorrência dessas conexões. A proposta permite separar o grafo em diferentes núcleos (áreas) a partir da avaliação de nós vizinhos que apresentam arestas de

maior peso. Para esse fim, exploraram conceito de similaridade para medir a porcentagem de nós iguais entre dois núcleos de diferentes grafos, tornando possível reconhecer quais áreas permaneceram sem grandes mudanças; ou se houve alguma aglutinação ou separação destas.

[Liu et al. 2019] propuseram um modelo de grafo derivado e uma ferramenta que analisa a evolução de um *TKG*, o *EvolveKG*. O trabalho adiciona um *timestamp* a tripla *RDF* (sujeito, predicado, objeto), de forma que distingua as relações de cada outra iteração no tempo.

[Pernischová et al. 2019] estudaram uma solução para analisar o impacto de uma operação em um *TKG*, de forma a poupar recursos para realizar uma atualização. Avaliando a somatória e a média entre as ocorrências de nós vizinhos entre duas iterações de um grafo, um modelo de aprendizado de máquina é gerado, analisando o impacto de uma operação de modificação no grafo, tal qual: `move`, `merge`, `split`, `add` e `delete`, em uma medida de centralidade, como *degree centrality*, *closeness centrality* e *betweenness centrality*.

Buscando representações visuais para apresentar informações relevantes de um grafo, [Uchoa Maia Rodrigues et al. 2019] investigaram uma abordagem de codificação gráfica, na forma de uma imagem, para a visualização da evolução de diferentes métricas em um grafo. Exploraram um exemplo de uma análise de um jogo de futebol na criação do grafo; a posição dos jogadores no campo foi codificada como os nós e a possibilidade de passar a bola para um jogador do mesmo time foi expressada como as arestas. Isso foi computado para cada instante relevante do jogo então calculada a entropia de cada nó, que tem uma codificação equivalente em cores — quanto maior a entropia, mais saturada é a cor. Ao final do processo, repetido para cada grafo gerado, obtém-se uma representação gráfica que codifica o valor de entropia para cada nó a cada instante do tempo.

[Beck et al. 2017] realizaram uma meta-análise do estado da arte de representação de grafos dinâmicos estudando 162 publicações, entre 1992 a 2015. As visualizações são organizadas em duas categorias principais: as que utilizam algum recurso de animação para mostrar a mudança de cada passo do grafo de maneira fluida; e as que utilizam um mapeamento de tempo para espaço para representar as alterações, resultando em uma estrutura que se assemelha a uma linha do tempo.

A abordagem animada facilita a assimilação do usuário acerca das mudanças aplicadas no grafo quando a maior parte da estrutura desse se mantém estática (isto é, grande parte dos vértices não se movimenta) [Beck et al. 2017]. Isso se mostra uma barreira quando várias mudanças são aplicadas no grafo em um mesmo passo, visto que, além de grande parte da sua estrutura se movimentar, tudo é realizado ao mesmo tempo.

Técnicas para diminuir o impacto dessa dificuldade foram estudadas. Por exemplo, o agrupamento dos nós mais frequentemente vistos juntos em *clusters* e aplicando um guia visual extra de forma a identificar esse novo grupo, como um círculo ao redor dos vértices integrantes [Beck et al. 2017]. Para passos mais complexos ou mais numerosos, é preferida a abordagem de linha do tempo, na qual o tempo é mapeado diretamente do grafo, como adicionando a informação do tempo em uma codificação por cores nas arestas dos vértices. Isso expressa quais iterações essa tripla esteve presente, bem como a ordem das suas ocorrências.

Nossa proposta se direciona a visualizar TKGs gerados com base em diversas instâncias de KGs gerados a partir de textos no tempo. O *TKGEvolViewer* (Seção 3) apresenta diversas funcionalidades para a exploração visual de TGKs e a análise visual com base em métricas computadas a partir das estruturas do grafo.

3. TKGEvolViewer

Desenvolvemos um método de visualização de TKGs, permitindo a utilização tanto por um usuário especialista, quanto para aqueles menos experientes. Isto é possível através de um fluxo de utilização bem definido, e também da possibilidade de livre exploração interativa dos elementos do TKG. Esses aspectos foram considerados no design e implementação da ferramenta de *software* **TKGEvolViewer**[1].

O *TKGEvolViewer* considera como entrada o TKG propriamente dito e medidas de centralidades previamente extraídas desse grafo. Dados de entrada são gerados pela ferramenta *TKGAnalyzer* [Rossanez et al. 2020a]. Medidas de centralidades ajudam a responder questões acerca do grafo. Quando presentes, essas métricas são utilizadas no auxílio na visualização do TKG, principalmente por usuários menos experientes.

Utilizando elementos gráficos comuns, o *TKGEvolViewer* apresenta um visual minimalista e módulos de auxílio para a obtenção de informações. Requisitos de recuperação simplificada de dados foram considerados. Implementamos meios de se obter uma exploração completa dos conjuntos de dados (os TKGs). A ferramenta permite a exploração de diversas iterações entre os dados, em diferentes instâncias do tempo.

3.1. Fluxo de interação e funcionalidades principais

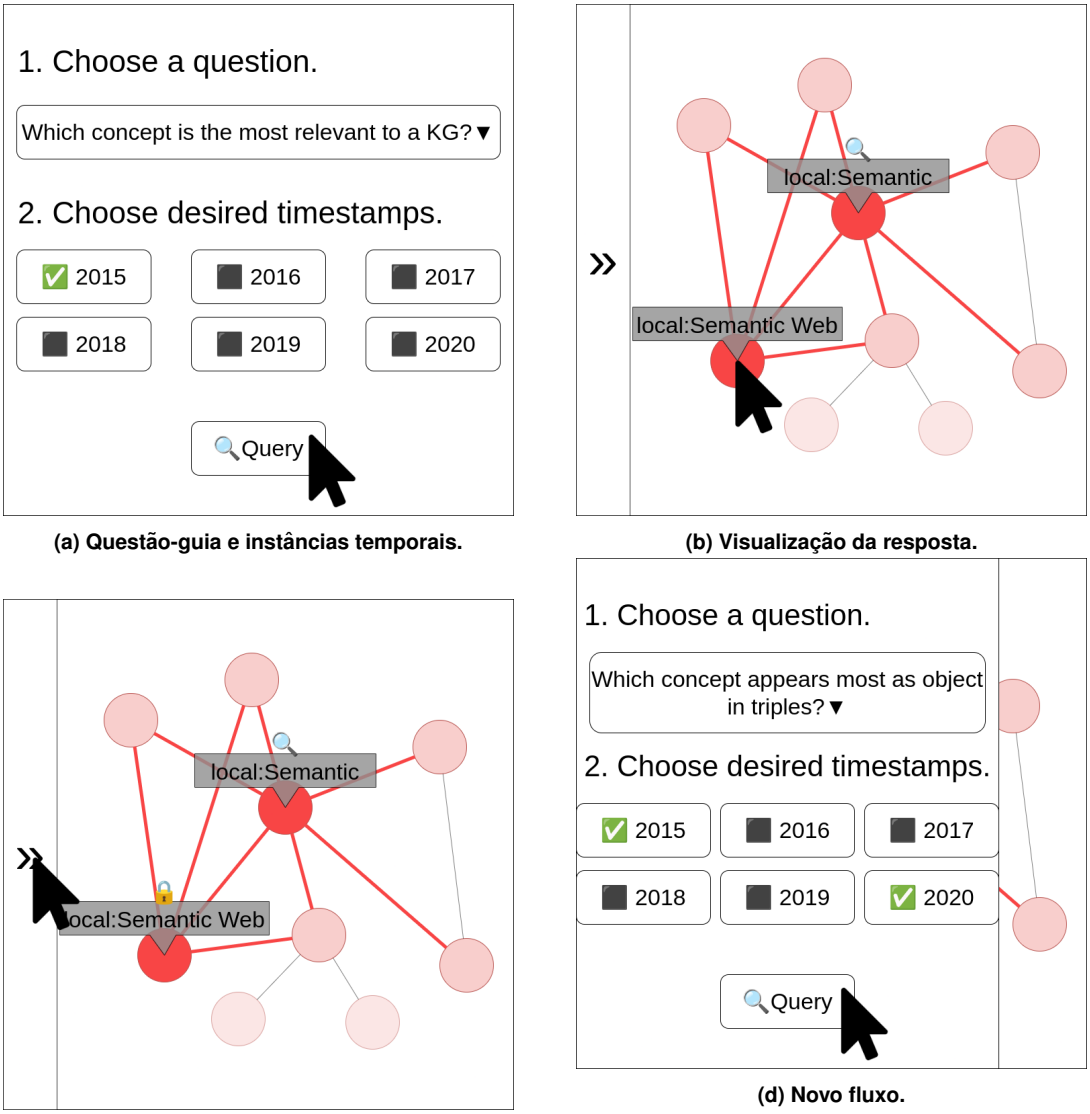
Ao se iniciar o *TKGEvolViewer* e ser carregado o conjunto de dados de entrada, a ferramenta apresenta ao usuário um *Wizard*, que implementa o fluxo de interação. Como ponto de partida, há uma questão-guia, que é respondida utilizando-se as métricas de centralidade previamente fornecidas. O objetivo do *Wizard* é guiar usuários na construção de análise sobre o TKG disponível. A partir da questão-guia, o *Wizard* conduz o fluxo de interação em um caminho que vai restringindo com o acréscimo de preferências sequenciais, de maneira a conduzir o usuário até o ponto visual que representa o conceito ou informação buscada. A Figura 1a apresenta a primeira tela do *wizard*, que permite escolher a questão-guia.

Questões-guia direcionam o usuário através das informações que podem ser obtidas de forma mais direta. Além das perguntas, a ferramenta apresenta um conjunto de instâncias temporais disponíveis junto aos dados de entrada. O usuário é direcionado a um ambiente que destaca os conceitos que respondem a questão-guia. Figura 1b apresenta a tela em que permite ao usuário navegar interativamente por conceitos de seu interesse.

Como próximo passo, a solução permite ao usuário iniciar um novo fluxo de busca, ao clicar no menu lateral (*cf.*, Figura 1c). A proposta permite ao usuário manter visível durante o novo fluxo, a resposta à questão inicial, como ilustrado pelo ícone de cadeado exibido na Figura 1c. O resultado da ação de clique é ilustrada na Figura 1d.

O *TKGEvolViewer* permite a usuários mais experientes uma interação que dispensa o *wizard*. Esses usuários podem navegar interativamente, através de busca e seleção

[1] `https://gitlab.ic.unicamp.br/ra124136/TKGEvolViewer/` (Acesso: Nov/2021)



(a) Questão-guia e instâncias temporais.

(b) Visualização da resposta.

(c) Revisitação à questão-guia.

(d) Novo fluxo.

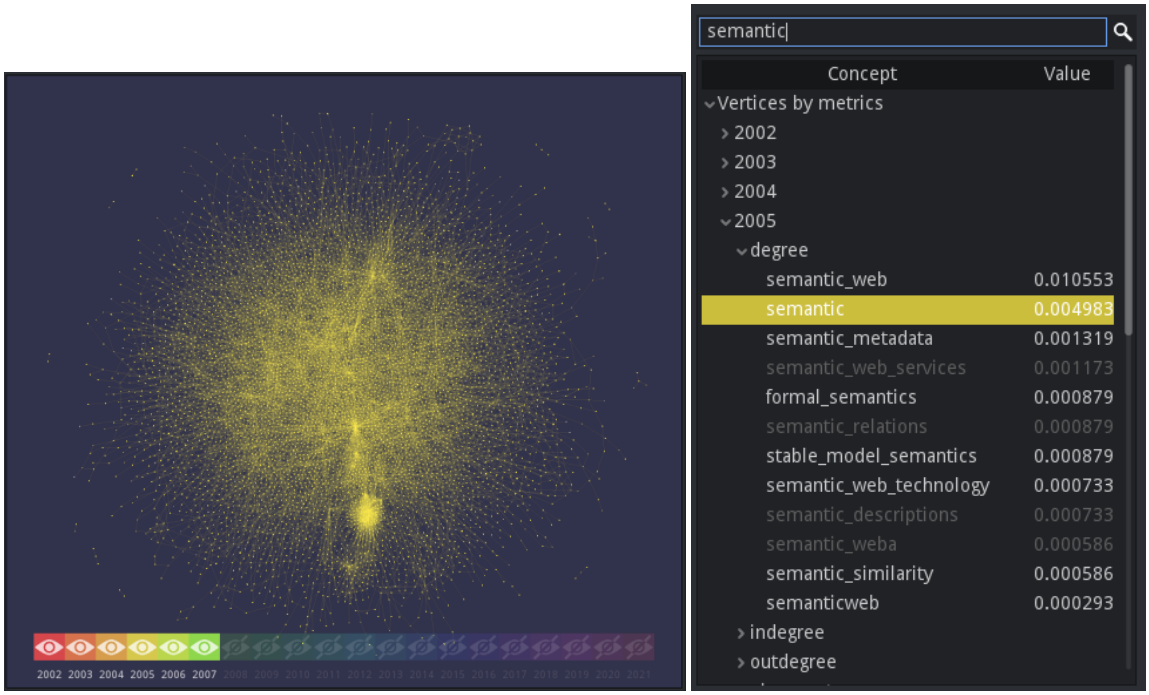
Figura 1. *Storyboarding* ilustrando o fluxo de utilização do *TKGEvolViewer*.

de conceitos diretamente na tela de visualização do TKG (*cf.*, Figura 2a). Figura 2a apresenta o grafo com uma linha do tempo, que indica as instâncias temporais disponíveis. Esta funcionalidade do *TKGEvolViewer* permite ao usuário explorar diretamente métricas de centralidade, através da *metrics tree* (*cf.*, Figura 2b).

3.2. Visão Arquitetural e Aspectos de implementação

A Figura 3 apresenta a visão arquitetural do *TKGEvolViewer*, que é composto por quatro módulos principais:

1. **GUI (Graphical User Interface)**: responsável pela interação com o usuário, exibição dos resultados e ativação dos módulos responsáveis pelas outras partes do sistema.
2. **TKGFactory**: responsável pela interpretação dos dados de entrada (TKGs), e integração com as bibliotecas externas.



(a) Visualização do grafo e instâncias temporais disponíveis. **(b) Busca de conceitos na *Metrics tree*.**

Figura 2. Funcionalidades do *TKGEvolViewer*.

3. **GDNTKG (Godot Native TKG)**: executa algoritmos de posicionamento de vértices em grafos do *igraph*.
4. **TKG e KG**: armazena dados de grafos criados pelo módulo *GDNTKG*.

O *TKGEvolViewer* foi desenvolvido considerando requisitos de desempenho pois o número de vértices em diversas iterações de um grafo pode alcançar dezenas de milhares. Por esta razão, foi escolhida uma plataforma enxuta e de fácil utilização, a *Godot Game Engine*[2]. Bibliotecas adicionais otimizadas para lidar com uma grande quantidade de dados e operações complexas, como a *igraph*[3], a *boost*[4] e a *annoy*[5] também foram utilizadas na implementação.

4. Estudo de caso

Objetivando avaliar o *TKGEvolViewer* de maneira qualitativa, conduzimos um estudo de caso baseando-se em um TKG real, obtido através de textos temporais em língua natural. Nesta seção, descrevemos como os dados de entrada foram obtidos, e ilustramos sua visualização pelo *TKGEvolViewer*.

4.1. Contexto

O TKG utilizado em nosso estudo de caso é baseado na conferência anual ISWC (*International Semantic Web Conference*[6]), da área de web semântica.

[2]`https://godotengine.org/` (Acesso: Dez/2020)
[3]`https://igraph.org/` (Acesso: Jul/2021)
[4]`http://boost.org/` (Acesso: Jul/2021)
[5]`https://github.com/spotify/annoy/` (Acesso: Nov/2021)
[6]`https://link.springer.com/conference/semweb` (Acesso: Nov/2021).

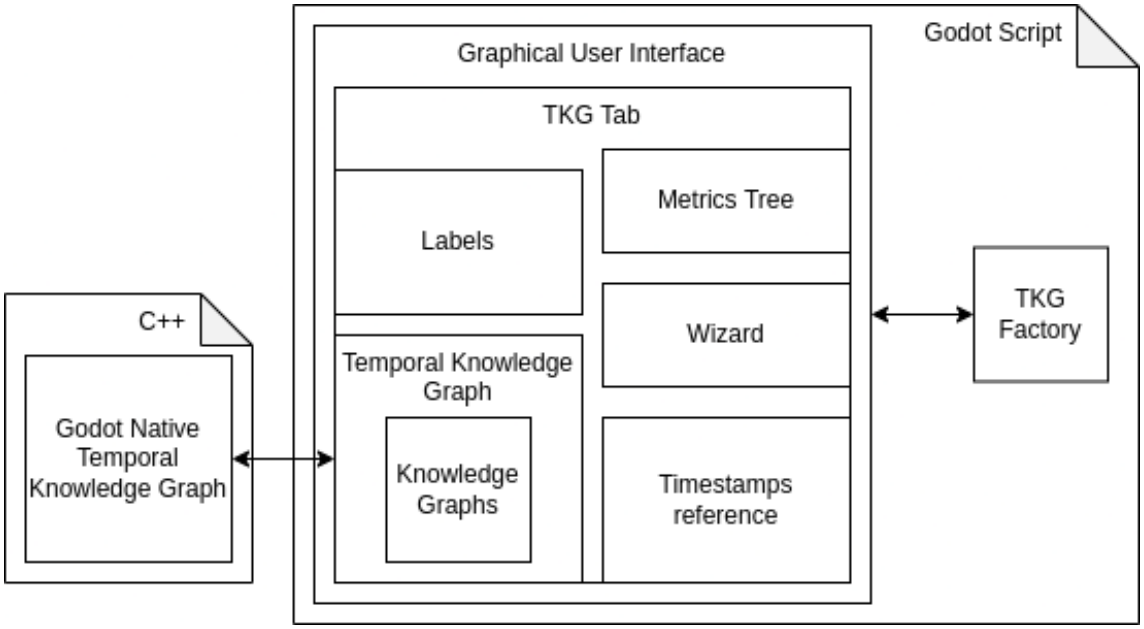


Figura 3. Diagrama arquitetural do *TKGEvolViewer*.

O referido TKG foi gerado com a ferramenta KGen [Rossanez et al. 2020b], utilizando os resumos de todos os artigos obtidos da trilha de pesquisa da conferência, entre os anos de 2002 e 2021. A ferramenta gera KGs utilizando técnicas de processamento de língua natural para a identificação e extração de conceitos e relacionamentos em textos não-estruturados, em triplas (sujeito, predicado, objeto). A ferramenta efetua ligações dos conceitos extraídos com classes de ontologias computacionais. Para complementar a entrada esperada do *TKGEvolViewer*, extraímos métricas de centralidades do TKG gerado. Para esse fim, utilizamos a ferramenta TKGAnalyzer [Rossanez et al. 2020a].

Entre os anos de 2002 e 2021, o TKG possui cerca de 89 mil conceitos, aproximadamente 2 mil mapeamentos em classes de ontologias e mais de 230 mil relacionamentos. O arquivo de entrada conta com 12 métricas de centralidade disponíveis para análise.

4.2. Visualização

Ilustramos um exemplo de execução com algumas ações específicas do usuário. A ferramenta apresenta inicialmente o *Wizard* (Figura 4).

O usuário prossegue selecionando a questão-guia apresentada — *Qual é o conceito de maior importância em cada edição da conferência?* A resposta à questão em específico é baseada na métrica *closeness centrality*, buscada pela ferramenta de maneira transparente ao usuário. O usuário seleciona o intervalo temporal compreendendo as edições da conferência ISWC de 2002 até 2007. Figura 5 apresenta a visualização que representa a resposta para a questão-guia e o intervalo temporal selecionado. O usuário pode selecionar o conceito relacionado à edição específica, selecionando o ano correspondente na barra temporal na parte inferior da Figura 5 (no caso do exemplo, o ano selecionado é 2005).

O usuário inicia um novo fluxo de execução no *Wizard*, com a mesma questão-guia do fluxo anterior, porém selecionando apenas a edição de 2002. A janela de visualização

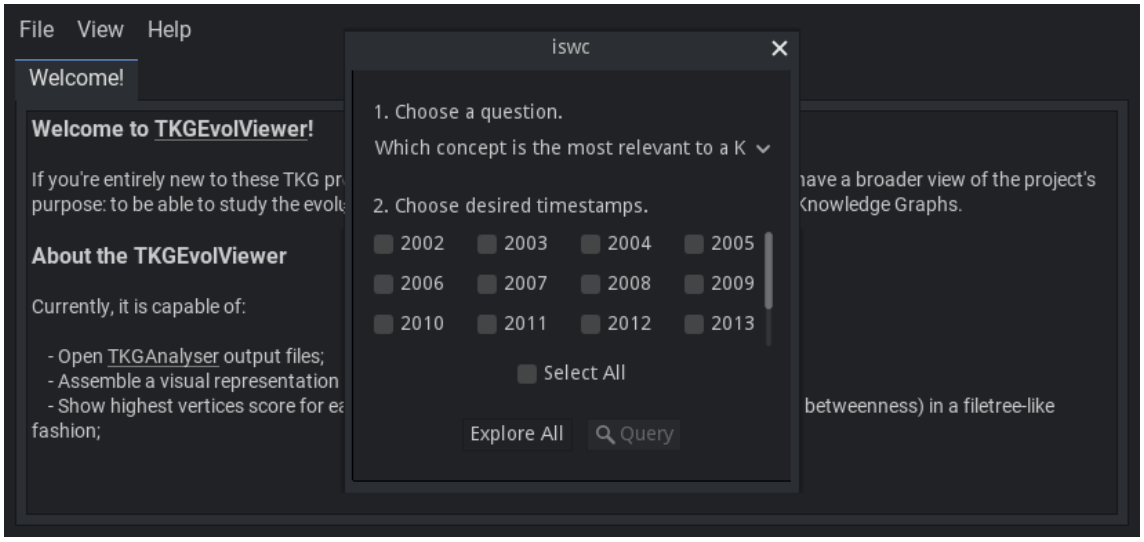


Figura 4. Tela inicial – o *Wizard*.

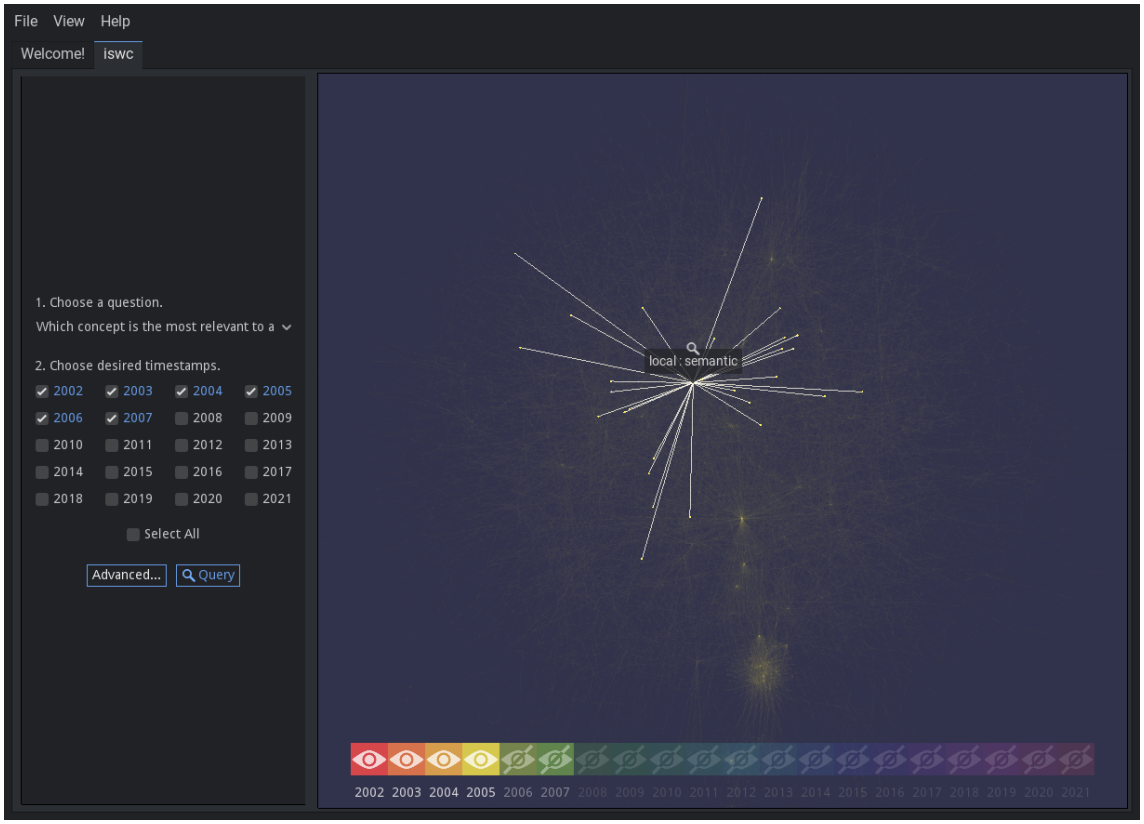


Figura 5. Conceitos mais importantes nas edições de 2002 até 2007. Visualização da edição de 2005.

do grafo apresenta o conceito *software*, que apresenta o maior valor para a métrica *closeness centrality* no ano de 2002. Nesse momento o usuário considera o conceito *software* na visualização do grafo. Figura 6 apresenta um cenário em que o usuário passa então a explorar a *Metrics tree*, selecionando o conceito *RDF*, que é apresentado na visualização do grafo.

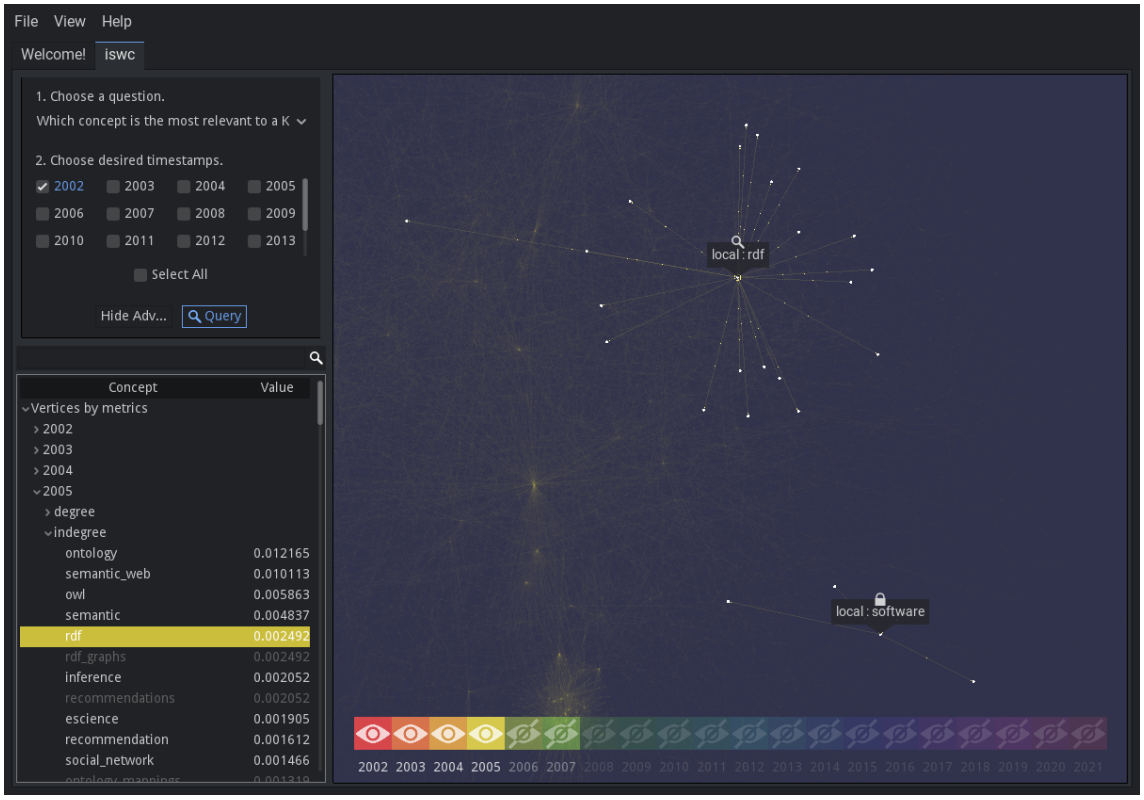


Figura 6. Exploração da *metrics tree* após utilização inicial do *wizard*.

5. Discussão

A visualização de grafos dinâmicos ainda é um problema em aberto, com algumas abordagens estudadas, como: a) a representação do tempo em um mapeamento espacial; b) em forma de animação; c) a codificação visual em um esquema de mapa de calor. Realçamos a relevância em se permitir armazenar o grafo de forma completa enquanto mantém a flexibilidade da representação, não necessitando realizar mudanças complexas como utilizar diversas linguagens de programação ou incorporar diferentes bibliotecas de auxílio visual para mudar a sua representação.

Nossa proposta neste estudo, a ferramenta *TKGEvolViewer* apresentou uma base sólida que supre as carências do usuário iniciante através do *Wizard*, apresentando um caminho guiado acerca das capacidades do *TKGEvolViewer*, disponibilizando perguntas selecionadas especificamente para o tema de evolução de grafos de conhecimento.

Nossa proposta visou igualmente satisfazer usuários mais experientes, possibilitando que eles interajam com os grafos de uma maneira exploratória. A funcionalidade de *Metrics tree* permite relacionar a pontuação dos conceitos nas métricas medidas com a posição e suas conexões nas iterações desejadas.

A concepção arquitetura da ferramenta *TKGEvolViewer* permite maleabilidade e modularidade para avanços em novas frentes e funcionalidades. Ela permite a adição de módulos que auxiliem a experimentação de novos métodos de obtenção de informações sobre os grafos. Podemos incluir novos módulos para representações gráficas adicionais.

Dentre os trabalhos futuros, consideramos implementar novas funcionalidades

para refinamento do *Wizard*, permitindo que o usuário tenha mais caminhos a explorar em sua navegação. Além das questões pré-definidas, esperamos permitir que o usuário construa suas próprias questões e auxilie a ferramenta em novos caminhos e análises. Um outro ponto a ser explorado é a inclusão de visualizações baseadas em realidade virtual, inserindo o usuário de maneira imersiva no contexto do TKG.

6. Conclusão

KGs são muito utilizados na representação e recuperação de conhecimento em um dado domínio de aplicação. Essas estruturas podem se tornar incrivelmente complexas, especialmente em se considerando sua representação gráfica. A quantidade de nós e arestas pode se tornar extremamente grande. TKGs são recursos relevantes na representação de como o conhecimento evolui ao longo do tempo em um domínio de aplicação. Tal particularidade é especialmente observada quando se trata de TKGs. Esse contexto demanda soluções que permitam a visualização e exploração desses artefatos. Nesta investigação apresentamos o *TKGEvolViewer* que auxilia na exploração e visualização de grafos de conhecimento temporais. Nossa ferramenta de software desenvolvida prioriza a experiência de uso e desempenho na execução visando tratar altas quantidade de nós e arestas. Estudos futuros envolvem avaliação experimental aprofundada e refinamentos em componentes chaves da ferramenta.

Referências

Arnaout, H. and Elbassuoni, S. (2018). Effective searching of rdf knowledge graphs. *Journal of Web Semantics*, 48:66–84.

Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2017). A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159.

Chen, F., Doan, A., Yang, J., and Ramakrishnan, R. (2008). Efficient information extraction over evolving text data. In *2008 IEEE 24th International Conference on Data Engineering*, pages 943–952.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., and Zimmermann, A. (2021). Knowledge graphs. *ACM Comput. Surv.*, 54(4).

Kim, H. (2017). Towards a sales assistant using a product knowledge graph. *Journal of Web Semantics*, 46-47:14–19.

Liu, J., Zhang, Q., Fu, L., Wang, X., and Lu, S. (2019). Evolving knowledge graphs. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2260–2268.

Pernischová, R., Dell'Aglio, D., Horridge, M., Baumgartner, M., and Bernstein, A. (2019). Toward predicting impact of changes in evolving knowledge graphs. In *SEMWEB*.

Pomp, A., Kraus, V., Poth, L., and Meisen, T. (2020). Semantic concept recommendation for continuously evolving knowledge graphs. In Filipe, J., Śmiałek, M., Brodsky, A., and Hammoudi, S., editors, *Enterprise Information Systems*, pages 361–385, Cham. Springer International Publishing.

Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., and Bogaard, T. (2016). Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37-38:132–151.

Rossanez, A. and Dos Reis, J. C. (2019). Generating knowledge graphs from scientific literature of degenerative diseases. In *Proceedings of the 4th International Workshop on Semantics-Powered Data Mining and Analytics*, SEPDA 2019, pages 12–23.

Rossanez, A., Dos Reis, J. C., and Torres, R. d. S. (2020a). Representing scientific literature evolution via temporal knowledge graphs. In *Proceedings of the 6th Workshop on Managing the Evolution and Preservation of the Data Web, MEPDaW 2020*, pages 33–42.

Rossanez, A., Dos Reis, J. C., Torres, R. d. S., and De Ribaupierre, H. (2020b). KGen: A knowledge graph generator from biomedical scientific literature. *BMC Medical Informatics and Decision Making*, 20(S4).

Singh, K., Lytra, I., Radhakrishna, A. S., Shekarpour, S., Vidal, M.-E., and Lehmann, J. (2020). No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *Journal of Web Semantics*, 65:100594.

Tosi, M. D. L. and dos Reis, J. C. (2022). Understanding the evolution of a scientific field by clustering and visualizing knowledge graphs. *Journal of Information Science*, 48(1):71–89.

Uchoa Maia Rodrigues, D. C., Moura, F. A., Cunha, S. A., and da S. Torres, R. (2019). Graph visual rhythms in temporal network analyses. *Graphical Models*, 103:101021.