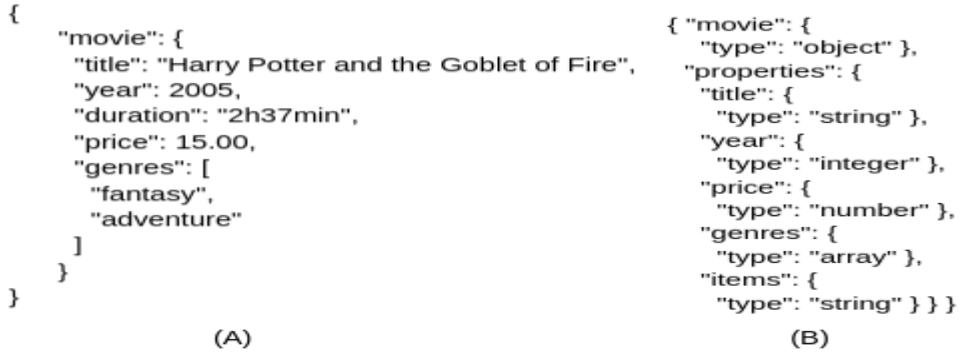
EXTRAÇÃO DE ESQUEMAS DE FONTES HETEROGÊNEAS SEMIESTRUTURADAS: UMA ABORDAGEM PROBABILÍSTICA

NATÁLIA BANHARA[1,2], GEOMAR SCHREINER[3], DENIO DUARTE[2,4]

1 Introdução

JSON Schema é um formato de esquema amplamente utilizado para descrever a estrutura das coleções de documentos JSON. Com essa estrutura, podemos verificar se os dados respeitam as restrições necessárias e, portanto, operações, como consulta e armazenamento, podem ser otimizadas (Bouchou e Duarte, 2007). Essa otimização é essencial principalmente considerando que coleções de documentos JSON são utilizadas para armazenar um grande volume de dados.

Figura 1. Um documento JSON (A) e seu esquema (B).



Como é possível observar na Figura 1(A), o formato JSON refere-se a pares chave-valor. Segundo Frozza et al. (Frozza et al. 2018), uma chave é uma sequência de caracteres sempre seguida por ":". Um valor, no entanto, pode se apresentar como uma estrutura complexa ou atômica.

No exemplo, é possível observar as classificações numéricas e sequência de caracteres (amplamente conhecida por string), pertencentes à definição de tipos atômicos. Além dessas, também é possível encontrar booleanos e nulos. Objetos e listas de valores (normalmente

[1] Graduação em Ciência da Computação, Universidade Federal da Fronteira Sul, *campus Chapecó*, contato: natalia.banhara@outlook.com
[2] Grupo de Pesquisa: Inovação e desenvolvimento tecnológico.
[3] Titulação acadêmica: Doutor, instituição Universidade Federal da Fronteira Sul.
[4] Titulação acadêmica: Doutor, instituição Universidade Federal da Fronteira Sul, **Orientador**.

chamados por arrays) são tipos complexos. Um objeto contém um conjunto desordenado de pares chave-valor, enquanto um array, uma coleção ordenada de valores (Frozza et al. 2018). Ambos representados na figura acima, respectivamente pelas chaves *movie* e *genres*. A Figura 1(B) apresenta um esquema que o documento em 1(A) respeita.

2 Objetivos

O objetivo deste trabalho é desenvolver uma ferramenta voltada à extração de um esquema com *tagged unions* e *enums*.

3 Metodologia

Trabalhos publicados entre 2017 e 2022 foram selecionados a partir da procura com a *string "JSON AND ("schema extraction" OR "schema inference" OR "schema discovery")"* na plataforma Google Scholar. A partir disso, 494 trabalhos foram retornados e sete foram selecionados, segundo: (*i*) relevância com o tema, (*ii*) qualidade do meio de publicação, e (*iii*) quantidade de citações.

Analisando os sete trabalhos foi possível identificar o estado da arte e comparou-se este com o resultado que (Imhof et al. 2017) apresentaram no que diz respeito ao período de 2012 à 2016.

Ademais, designou-se o uso da linguagem de programação *C++* para o desenvolvimento da ferramenta. Essa possui uma estrutura de dados chamada grafo em que os dados extraídos a partir de um documento JSON são estruturados na memória. Um grafo é definido pela tupla G = (V, E), onde V representa um vértice e E, uma aresta. Representando as chaves como V e a conexão (ou relacionamento) entre chaves como E, é possível visualizar um documento JSON.

4 Resultados e Discussão

A análise dos trabalhos deu forma ao artigo "Extração de Esquemas de Documentos JSON: O que há de novo?" (Banhara *et al.* 2023). Nesse artigo, o estudo de (Klessinger et al. 2022) voltou-se à detecção de *tagged unions* através de heurísticas. *Tagged unions* refere-se à dependência entre irmãos. Dado uma chave c1 com valor v1, c2 ocorre. Ou seja, v1 → c2.

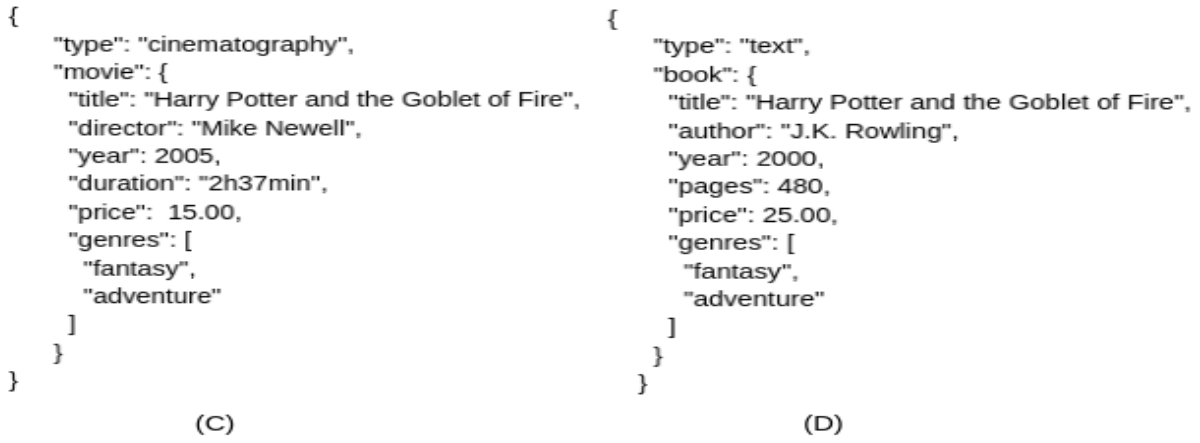
Enums são um conjunto de valores que uma dada chave pode assumir. O tipo *tagged unions*, por sua vez, são chaves do tipo enum que, conforme o valor, podem ter irmãos

diferentes.

Na Figura 2 é possível observar as definições mencionadas. Dependendo do valor de *type*, tem-se uma determinada estrutura no seu irmão imediato (comportamento de *tagged union*). Assim, observando a Figura 2, quando o tipo de informação é cinematografia 2(C), tem-se dados sobre filmes. Já quando é texto, dados específicos de livros 2(D)

A própria chave *type*, neste caso, exemplifica um *enum.* Ela só pode assumir dois valores: *cinematography* e *text*. Qualquer outro não faria sentido quanto ao contexto.

Figura 2: Exemplo de *tagged union* e *enum*.



Detectando *enums* e *tagged unions* é possível impedir que valores não relacionados sejam informados para uma determinada chave e a relação entre duas chaves possa ser inferida, respectivamente, refinando a extração do esquema JSON.

A ferramenta foi desenvolvida utilizando a estrutura de dados conhecida como grafo. Esse grafo é uma tupla G = (V, E) no qual V (vértice) é uma tupla *v = <l, T, c, isEnum, isTU>*, onde *l* é o nome da chave; *T* uma tupla *<t1: occ, ..., tn: occn> (t1* representa um tipo de *l* e *occ*, a quantidade de vezes que *l* aparece com dado tipo); *c* é a quantidade de vezes que a chave ocorre; *isEnum* identifica se os valores que a chave obteve representam uma enumeração; e *isTU* identifica se *l* origina uma *tagged union*.

Uma enumeração é representada no grafo, como explicado, a partir de *isEnum*, quando verdadeiro. Uma chave é dita *enum* caso a quantidade de valores únicos seja menor que um limite proposto. Quanto aos tipos, um *enum* só pode ser do tipo inteiro ou *string*. Esses precisam estar abaixo de um limite em relação à sua quantidade e comprimento respectivamente. Ainda, *enums* só ocorrem em chaves de tipo *array* ou atômico.

Tagged unions possuem os seguintes comportamentos. Uma chave só pode gerar uma

tagged union caso contenha uma enumeração. Além disso, a variação dos valores desta chave que originam uma outra, deve estar abaixo de um limiar.

Uma aresta (E) consiste na tupla $\varepsilon = <(vs, vt), rs, c\varepsilon, lv\varepsilon>$, onde *(vs, vt)* é uma tupla em que *vs* é o vértice de origem, enquanto *vt* o de destino; *rs* é o tipo de relacionamento entre os vértices que pode obter os valores pai ou irmão; $c\varepsilon$ diz respeito a quantidade de vezes que a aresta ocorreu e possibilita a inferência da obrigatoriedade (acima de um limiar) ou opcionalidade de uma aresta de *rs* pai, comparando com o valor *c* (de *vs*); e $lv\varepsilon$ é a lista de valores que ocorreram em *vs* quando irmão de *vt.*

Como resultado da ferramenta, um metamodelo foi criado a partir das propriedades extraídas. Como pode ser observado na Figura 3, a detecção de *tagged unions* é sinalizada por *$tag* e seguida pela descrição do que cada valor da enumeração de *type* gera. Caso nada seja gerado, o valor é listado como nulo. Como a chave *genres* se trata de uma enumeração, ela é identificada por *$enum*, listando os valores deste *enum*. Importante notar que caso *genres* fosse uma campo atômico, o seu tipo também estaria denotado. O símbolo *?* em *pages* exemplifica a opcionalidade.

Figura 3: Metamodelo com detecção de *tagged unions* e *enum*.

```
{
   "type"$tag: string {
      cinematography: "movie": {
         "title": string,
         "director": string,
         "year": integer,
         "duration": string,
         "price": double,
         "genres"$enum: [adventure, fantasy]
      }
      text: "book": {
         "title": string,
         "author": string,
         "year": integer,
         "pages"?: integer,
         "price": double,
         "genres"$enum: [adventure, fantasy]
      }
   }
}
```

5 Conclusão

Documentos JSON são amplamente utilizados em bancos com grande volume de dados. A extração de seus esquemas é essencial para que restrições e operações possam ser incorporadas de forma eficiente. A ferramenta proposta se encaixa neste contexto: permite que esquemas sejam extraídos e, consequentemente, otimizações sejam aplicadas no gerenciamento

de tais coleções.

Foram analisados sete trabalhos sobre o tema publicados de 2017 a 2022. Percebeu-se que nenhum dos trabalhos analisados incorpora *tagged unions* e *enums* nos esquemas extraídos. Assim, um grafo foi gerado para a proposição de um metamodelo. A partir desse metamodelo, é possível gerar um esquema JSON válido que descreve a coleção de documentos de entrada. É relevante mencionar que a ferramenta ainda está em desenvolvimento e mais experimentos precisam ser realizados.

Referências Bibliográficas

Banhara, Natália; Duarte, Denio; Schreiner, Geomar. Extração de Esquemas de Documentos JSON: O que há de Novo?. In: Escola Regional de Banco de Dados (ERBD). Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023 . p. 11-20.

DOI: https://doi.org/10.5753/erbd.2023.229421.

Bouchou, B. and Duarte, D. (2007). Assisting XML schema evolution that preserve validity. In Simpósio Brasileiro de Banco de Dados - SBBD, pages 270–284.

Frozza, A. A., dos Santos Mello, R., and da Costa, F. d. S. (2018). An approach for schema extraction of json and extended json document collections. In 2018 IEEE International Conference on Information Reuse and Integration (IRI), pages 356–363. IEEE.

Imhof, R., Frozza, A. A., and dos Santos Mello, R. (2017). Um survey sobre extração de esquemas de documentos json. In Anais da XIII Escola Regional de Banco de Dados. SBC.

Klessinger, S., Klettke, M., Störl, U., and Scherzinger, S. (2022). Extracting json schemas with tagged unions. *coordinates*, 30:10.

Palavras-chave: JSON; extração de esquema; grafo; probabilidade.

Nº de Registro no sistema Prisma: PES-2022-0312

Financiamento: Universidade Federal da Fronteira Sul