MODELAGEM COMPUTACIONAL PARA O PRÉ-PROCESSAMENTO DA BASIC TEXT PIPELINE COM MONGODB E SHARDING

Matheus Sardeli MALHEIROS
Graduado em Sistemas para Internet
IFSP/Câmpus São João da Boa Vista

Gustavo Aurélio PRIETO
Mestre em Ciências da Computação - UFSCar
Vice-líder do Grupo de Pesquisas em Comunicação Científica aCOMTECe
Docente de Ciências da Computação
IFSP/São João da Boa Vista

Rosana Ferrareto Lourenço RODRIGUES
Doutora em Linguística e Língua Portuguesa/UNESP-Araraquara
Líder do Grupo de Pesquisas em Comunicação Científica aCOMTECe
Docente de Letras
IFSP/São João da Boa Vista
Docente do Mestrado ProfEPT
IFSP/Sertãozinho

RESUMO

A implementação do banco de dados MongoDB para armazenamento de informações é comum na área da Tecnologia e útil para a Educação. Esta pesquisa apresenta a modelagem do pré-processamento da Basic Text Pipeline para armazenamento e manipulação de resumos científicos implementado no MongoDB. É um banco de dados não-relacional, que não possui estrutura pré-definida e permite armazenar dados semi-estruturados e não-estruturados. Utiliza o método *Sharding* para dividi-lo entre servidores, que organizam e armazenam as informações em três áreas do conhecimento. A divisão possibilita verificar que as consultas são direcionadas ao respectivo *shard* quando utilizada a chave de fragmento, sendo necessário consultar apenas as informações deste para otimizar consultas e manipulação dos dados. Além disso, permite ao cientista de dados visualizar e analisar as informações para promover uma aplicação linguística, possibilitando ao linguista acompanhar o trabalho e utilizá-lo para o desenvolvimento de ferramentas tecnológicas linguísticas para o ensino de redação científica.

Palavras-chave: Banco de dados; NoSQL; MongoDB; *Sharding*; Basic Text Pipeline; Abstracts.

COMPUTATIONAL MODELING FOR BASIC TEXT PIPELINE PRE-PROCESSING WITH MONGODB AND SHARDING

ABSTRACT

The implementation of MongoDB database for information storage is common in the field of Technology and useful for Education. This research presents the Basic Text Pipeline pre-processing modeling for storing and manipulating scientific abstracts implemented in MongoDB. It is a non-relational database, which has no pre-defined structure and allows storing semi-structured and unstructured data. It uses the Sharding method to divide it among servers, which organize and store information in three knowledge fields. The division enables

to verify that the queries are directed to the respective shard when using the fragment key, and it is necessary to consult only its information to optimize queries and data manipulation. In addition, it allows the data scientist to visualize and analyze information to promote linguistic application, enabling the linguist to follow the work and use it to develop linguistic technological tools for teaching scientific writing.
Keywords: Database; NoSQL; MongoDB; Sharding; Basic Text Pipeline; Abstracts.

Introdução

Os bancos de dados estão cada vez mais presentes em nosso dia a dia, isso se deve ao desenvolvimento acelerado da internet, a partir da qual empresas utilizam aplicações sob forma de website, ao invés de sistemas fechados. Com isso, novas demandas foram aparecendo e as exigências de um banco de dados mudaram, já que os principais problemas e limitações encontrados no modelo relacional estão ligados à dificuldade de conciliar a demanda da escalabilidade juntamente ao tipo de modelo, devido ao alto crescimento de dados. Além disso, pode-se citar a utilização de banco de dados na educação, nos quais frequentemente precisa-se armazenar uma grande quantidade de informações e documentos, para posteriormente serem facilmente encontrados e manipulados, como é o caso de documentos e artigos científicos.

Nesse contexto, justifica-se este artigo pela necessidade de se disponibilizar e padronizar o armazenamento dos resumos científicos (*abstracts*) de alto impacto da Web of Science na etapa de pré-processamento da Basic Text Pipeline (BTP)[1], aplicando o método *Sharding*. Isso atende à demanda do Grupo de Pesquisas em Comunicação Científica aCOMTECe[2], em seu trabalho na educação de cientistas. Também possibilita a escalabilidade de futuramente a inserção de novos *abstracts* como dados da pipeline, já que a tecnologia NoSQL viabiliza a manipulação de grandes volumes de dados não estruturados e semiestruturados.

O objetivo geral é a implementação do banco de dados MongoDB utilizando Sharding para armazenamento de *abstracts* no pré-processamento da BTP, visando disponibilizar os dados para utilização na BTP, e assim, padronizar o local de armazenamento das informações. E, como objetivos específicos experimentar como o banco de dados do MongoDB armazena *abstracts* no pré-processamento da BTP,

[1]https://www.redhenlab.org/home/the-cognitive-core-research-topics-in-red-hen/the-barnyard/basic-text-pipeline
[2] https://acomtece.sbv.ifsp.edu.br/

verificar como o método *Sharding* do MongoDB divide a base de dados entre servidores e analisar e testar se os *abstracts* estão armazenados na base de dados.

A metodologia se insere na etapa de pré-processamento da implementação do banco de dados MongoDB, utilizando o método *Sharding* para armazenamento de dados semi-estruturados. Para isso foram levantados e coletados os dados a serem armazenados, verificadas a arquitetura e modelagem para configuração do banco de dados e aplicados os dados dentro da base de dados.

Fundamentação Teórica

Os dados são compreendidos como sequências de símbolos que possuem valor e podem ser armazenados em algum dispositivo, tais como um banco de dados ou documentos (SETZER; SILVA, 2005; SETZER, 1999, VAZ; 2000). Logo, são "fatos conhecidos que podem ser registrados e possuem significado implícito" (ELMASRI; NAVATHE, 2010, pp. 3), como por exemplo: O código de barras de uma mercadoria, o qual é um fato conhecido representado por símbolos e assim capaz de ser registrado.

Dados carregam junto de si alguns outros dados que têm a função de identificar ou documentar algum recurso, e, com isso, deve-se partir para a definição de metadados. Metadados podem ser compreendidos como uma informação que descreve um dado, ou seja, um dado que descreve o atributo de um recurso. Além disso, eles são responsáveis por ordenar, selecionar, localizar, avaliar e documentar objetos, já que apresenta uma descrição resumida e precisa a respeito do dado, com a finalidade de filtrar o acesso ao mesmo, como por exemplo o nome de um arquivo no computador, o título de um livro ou o código de barras de um produto (VAZ, 2000; ALVES; SOUZA. 2007). No caso da BTP, os *abstracts* possuem diversos metadados, tais como a sigla "TI" que representa o título e "PY" que é o ano de publicação.

Os dados e metadados, por sua vez, precisam ser armazenados de alguma maneira para que posteriormente possam ser acessados. Para isso, é necessário compreender o conceito de banco de dados.

Inicialmente, originado do termo inglês *Databanks*, em seguida trocado pela expressão *Database* (Base de Dados), o banco de dados pode ser compreendido como um local de armazenamento de dados. É um conjunto de dados operacionais persistentes que estão relacionados entre si de alguma maneira e estão organizados de maneira a

servir um conjunto de aplicações. Logo, o banco de dados representa algum aspecto do mundo real e é uma coleção de dados organizados de maneira lógica coerente e com algum significado inerente, como o exemplo de uma lista telefônica e um catálogo de DVDs (SETZER; SILVA, 2005; ELMASRI; NAVATHE, 2010, pp. 3; DATE, 2004, pp.6).

Para o gerenciamento do banco de dados e dos dados nele armazenado, existe o Sistema Gerenciador de Banco de Dados (SGBD), do inglês *Data Base Management System* (DBMS), que pode ser compreendido como o *software* utilizado para a administração do banco de dados, permitindo realizar a criação, alteração, inserção, buscas e exclusão dos dados. Além disso, o SGDB provê acesso rápido aos dados desejados. É um sistema responsável por armazenar as informações e permitir que os usuários realizem as buscas e atualizações dessas informações quando for solicitado (DATE, 2004; RAMAKRISHNAN; GEHRKE, 2007).

Como a estrutura do banco de dados está diretamente ligada com o modelo de dados, é necessário compreender que se trata de uma coleção de ferramentas conceituais que descrevem: os dados, a semântica dos dados e as restrições de consistência. Logo, sua finalidade é descrever o projeto de um banco de dados de maneira lógica, física e de *view* (SILBERSCHATZ et al, 2012).

Com base no modelo de dados, existem o Modelo Relacional (MR) e Modelo Não-Relacional (NoSQL, do inglês *Not Only* SQL). O MR pode ser compreendido como um modelo no qual o banco de dados é representado por um conjunto de relações, no qual uma relação equivale a uma tabela de valores e quando aplicado a terminologia MR, as colunas são consideradas atributos; as linhas, tuplas; e a tabela em si, relação (ELMASRI; NAVATHE, 2010). Por sua vez, o NoSQL pode ser compreendido como um modelo no qual o banco de dados pode armazenar dados semiestruturados e não-estruturados. Os dados não-estruturados podem ser compreendidos como os dados sem estrutura prévia nem possibilidade de agrupamento em tabelas, como vídeos, imagens e e-mail; e os dados semiestruturados são irregulares ou incompletos não necessariamente de acordo com um esquema. Documentos HTML e logs de *website* são compreensíveis por máquinas, mas não por seres humanos (INTEL, 2015; LÓSCIO et al, 2011).

Logo, o surgimento do NoSQL ocorreu a partir do crescimento exponencial da quantidade de dados gerados pelas pessoas nos últimos anos, devido à popularização e acesso à Internet, além do aumento significativo do acesso a dispositivos eletrônicos como, por exemplo, *smartphones* e *notebooks*. Com isso, foi necessária a criação de uma tecnologia que possibilitasse a manipulação de grandes volumes de dados não-estruturados e semiestruturados, juntamente às necessidades de disponibilidade e escalabilidade. Isso se deve ao fato de os principais problemas e limitações encontrados no modelo relacional estarem ligados à dificuldade de conciliar a demanda da escalabilidade juntamente ao tipo de modelo (LÓSCIO et al, 2011).

Os modelos de dados dos bancos de dados NoSQL podem ser divididos entre quatro grandes grupos: modelo chave/valor, modelo orientado a documentos, modelo orientado a grafos, modelo orientado a colunas (CATTELL, 2010).

A BTP tem como fonte de informação os *abstracts* coletados na Web of Science[3], que, por sua vez, são documentos. Verificou-se que o Modelo Orientado a Documentos é o que melhor se enquadra, já que suas características estão diretamente relacionadas aos arquivos que precisam ser armazenados, nos quais cada item armazenado é um documento e não utiliza qualquer tipo de estrutura pré-definida.

A partir da escolha do modelo orientado a documentos, foi decidido o banco de dados a ser utilizado, o MongoDB, já que suas características estão ligadas às necessidades da BTP, além de ser um dos mais recentes e utilizados.

O MongoDB é um SGBD NoSQL Orientado a Documentos lançado em 2009 e foi desenvolvido para aplicações *web* e também para infraestruturas de internet, já que ele permite o armazenamento de dados em coleções de documentos e também oferece a possibilidade de escalar o banco de dados e dividi-los entre computadores (BANKER, 2011). Com isso, os Modelos Orientados a Documentos são diferentes dos bancos de dados com MR, já que armazenam os documentos vagamente definidos, ao invés de armazená-los em estruturas rígidas, como as tabelas (CUNHA, 2011). Vários projetos usam esse SGBD como a SEGA, Telefônica e Adobe, além de possuírem diversos drivers disponíveis para as linguagens de programação C, C++, Java, Node.js, PHP, Python, Ruby, entre outras (DOCS.MONGODB, 2021).

[3] https://www.webofknowledge.com/

A persistência dos documentos no MongoDB é feita no formato BSON, cada documento é a unidade de dados armazenável e um conjunto de documentos está dentro de uma coleção. O formato BSON pode ser compreendido como uma representação binária de documentos JSON (*JavaScript Object Notation*). Esse por sua vez é muito utilizado em aplicações que usam API e por isso é bastante conhecido. O BSON possui uma estrutura para mapeamento dos dados no formato de chave e valor, em que cada chave é caracterizada como identificação do atributo e deve estar cercada por aspas e o seu referente valor deve estar após os dois pontos; as possibilidades de atribuição são uma cadeia de caracteres, um número ou até mesmo um outro documento JSON. Esse formato torna mais rápida, leve e eficiente a análise de dados (HOWS et al, 2015).

O MongoDB possui outras duas características principais: o *Sharding* e o GridFS. O *Sharding* pode ser entendido como o método que o MongoDB utiliza para trabalhar em modo distribuído. A função desse método é dividir a grande quantidade de dados entre os computadores, sem perder desempenho e, com isso, manter os dados balanceados entre eles (SHARDING, 2021). Por sua vez, o GridFS é definido como o método que o MongoDB utiliza para o armazenamento e recuperação de arquivos que sejam maiores que o limite de tamanho do documento BSON de 16MB, como, por exemplo, um arquivo de imagem, de vídeo ou um grande documento de texto (DOCS.MONGODB, 2021).

Como a BTP utiliza arquivos de texto, o MongoDB é necessário, pois ele possibilita armazenamento dos dados em documentos vagamente definidos, além de possibilitar o uso do método *Sharding*, o que proporciona a escalabilidade na inserção de novos *abstracts* como dados da BTP. A tecnologia NoSQL viabiliza a manipulação de grandes volumes de dados não-estruturados e semiestruturados, além de oferecer, através do método GridFS, o armazenamento de arquivos independente do seu tamanho.

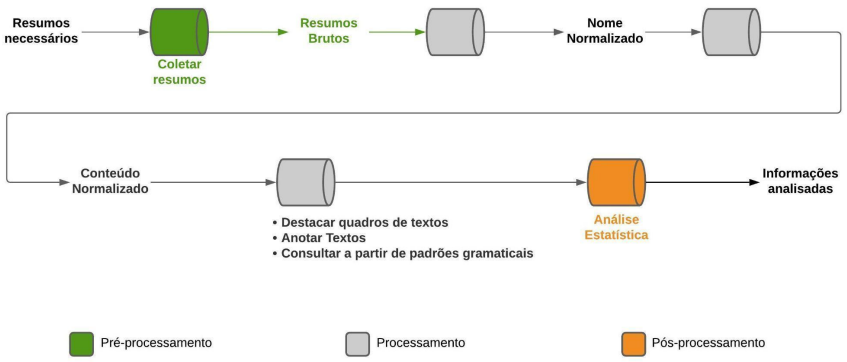
Metodologias

A Basic Text Pipeline (BTP) é um projeto do Laboratório Red Hen do Departamento de Ciências Cognitivas (CogSci) da Case Western Reserve University (CWRU) em Cleveland/OH, EUA. Esse processo, que faz parte do Projeto “Padrões Narrativos e Figurativos do texto científico” do aCOMTECe, desenvolvido em 2018, consistiu em bases teóricas e tecnológicas para a pesquisa em Comunicação Científica.

O objetivo foi analisar a semântica das construções de integração conceptual em *abstracts*, a partir da análise e manipulação de textos em língua natural da língua inglesa em prosa. A contribuição desse estudo é aplicar os resultados dos dados de análise na produção de ferramentas tecnológicas linguísticas para o ensino de redação científica.

Pode-se entender o processo da BTP em três fases: pré-processamento, processamento e pós-processamento. A fase de pré-processamento é a etapa de coleta dos resumos na base de dados científica, sua compilação e tratamento. O processamento é a fase onde são realizadas as manipulações dos arquivos, com o intuito de normalizar os arquivos, seus nomes e conteúdo; destacar quadros de textos; anotar textos e realizar consultas a partir de padrões gramaticais. A terceira e última fase, o pós-processamento, trata-se da análise estatística das informações obtidas do processamento (Figura 1). O foco deste trabalho é implementar o banco de dados MongoDB, utilizando o método *Sharding*, para armazenamento dos arquivos no pré-processamento.

Figura 1: Visão parcial do fluxo das fases da BTP



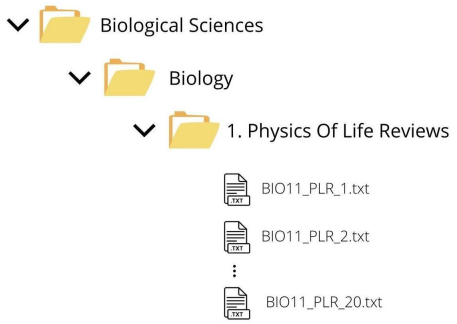
Fonte: BTP (2020)

Tendo em vista que o trabalho foi baseado no problema de pesquisa da BTP, primeiramente foram recuperados os formatos dos dados dos resumos já coletados para que se pudesse realizar a coleta dos dados. A partir da identificação dos dados já armazenados, foi feito o levantamento dos *abstracts* a serem coletados para completar o *corpus* da BTP, para que cada área do conhecimento tivesse o mesmo número de *abstracts* para o banco de dados.

O conjunto de dados que estava sendo trabalhado inicialmente na BTP era composto por 1.000 *abstracts*, os quais foram coletados manualmente pela linguista líder do Grupo de pesquisas aCOMTECe. Entretanto, emergiu a necessidade de igualar os *abstracts* entre as três áreas do conhecimento: Ciências Biológicas (*Biological Sciences*), Ciências Exatas (*Hard Sciences*) e Ciências Sociais e Humanas (*Social & Human Sciences*). Para atingir o mesmo número de *abstracts* entre as áreas, foram levantados números estatísticos apontando quantos resumos dentre os 1.000 eram de cada área e posteriormente, quantos resumos faltavam para completar a equalização com 1.800 resumos.

Após a definição dos arquivos a serem coletados, foram codificados os nomes dos resumos para fins de padronização quando da busca dos arquivos. Esse metadado é composto pela abreviação da disciplina, o número entre um e dezoito (número referente a disciplina entre as dezoito existentes no conjunto de dados da BTP, com seis disciplinas por área do conhecimento), sigla do periódico e o número referente ao resumo daquele periódico. Por fim, foram definidas a organização e nomeação das pastas para armazenar resumos da BTP. Com isso, as pastas ficaram organizadas seguindo a hierarquia: Área do conhecimento, posteriormente uma pasta para cada disciplina, dentro de cada disciplina uma pasta para cada periódico e dentro de cada periódico os *abstracts* coletados (Figura 2).

Figura 2: Taxonomia por áreas e disciplinas



Fonte: BTP (2020)

Posteriormente à coleta dos dados, foi analisada e elaborada a arquitetura e modelagem do banco de dados para armazenar os arquivos do pré-processamento. A

elaboração foi realizada a partir das características do MongoDB, que é SGDB consolidado com a abordagem não-relacional mais robusta, escolhido por possuir uma estrutura adequada ao armazenamento de informações dos *abstracts*. Além disso, possui grande poder de processamento paralelo, alta performance na recuperação dos dados, grande escalabilidade através do método *Sharding* e a especificação GridFS, a qual permite armazenar arquivos no banco independentemente do tamanho.

Partindo das características do MongoDB e do fato de que os dados são armazenados em coleções e documentos, foi definida a estrutura com três *shards*, responsáveis por armazenar os *abstracts* da respectiva área do conhecimento, com um para cada uma: Ciências Biológicas (*Biological Sciences*), Ciências Exatas (*Hard Sciences*) e Ciências Sociais e Humanas (*Social & Human Sciences*). Os três *shards* são responsáveis por dividir uma única coleção de pré-processamento, ficando cada *shard* encarregado por armazenar os dados e documentos de cada área.

Com base na estruturação das coleções e dos *shards,* foram definidas as informações dos documentos BSON, em que cada documento dentro da coleção pré-processamento contém as informações referentes a um resumo científico (ver Tabela 1).

Tabela 1: Definição das chaves e valores do documento BSON[4]

Chave	Descrição valor	Exemplo
TitleFileRaw	Título do Arquivo do Resumo Científico	BIO11_PLR_5
...	...	…
File	Conteúdo do Arquivo do Resumo Científico em sua totalidade (Autor, Título, Fonte, Resumo)	{“$binary”: “77u/Rk4gQ2xhcml2Y….”, “$type”: “0”}

Fonte: Autores (2020)

Além da definição das informações a serem armazenadas, foram definidos os campos *Field, Discipline* e *Journal* para a criação dos índices, já que estes suportam a execução eficiente de consultas no MongoDB. Sem índices, o MongoDB deve executar uma varredura de todos os documentos em uma coleção, para selecionar os documentos

[4] Tabela completa disponível em https://drive.google.com/file/d/142A6_OzgXkZcFufqSV_biR8OhUyRpRnX/view?usp=sharing

que correspondem à instrução de consulta. Se existir um índice apropriado para uma consulta, o MongoDB pode usar o índice para limitar o número de documentos que deve inspecionar (DOCS.MONGODB, 2021).

Em seguida, foram definidas as ferramentas e tecnologias para a configuração do banco de dados, para a inserção dos dados na base de dados e para o desenvolvimento da aplicação responsável por apresentar os dados armazenados. Para a etapa de configuração do banco de dados e do *Sharding*, foi utilizado o terminal do Ubuntu e também o MongoDB Compass. Para a etapa de desenvolvimento do algoritmo de inserção das informações no banco de dados, foi utilizada a linguagem de programação Python e o ambiente de programação PyCharm. Para a etapa de desenvolvimento da aplicação, foram utilizados PHP, Bootstrap, HTML, Javascript, PHPStorm e Xampp.

Com base nas ferramentas, tecnologias e linguagens de programação, partiu-se para a configuração do banco de dados MongoDB aplicando o método *Sharding* e o desenvolvimento do algoritmo para a inserção das informações dos *abstracts*.

Para a configuração do banco aplicando o *Sharding*, foi utilizado o terminal do Ubuntu e foram utilizados comandos para desabilitar o firewall do sistema operacional; criar e configurar pastas para cada um dos três *shards* e uma para o servidor de configuração; configurar a inicialização da replicação; configurar cada um dos *shards*; criar a base de dados “BasicTextPipeline”; definir a *shard key* (chave de fragmento responsável por determinar a distribuição dos documentos da coleção entre os *shards* do *cluster*/banco de dados); criar e configurar o compartilhamento da coleção “pre_processamento”; configurar as *shard zones*, responsáveis pelo agrupamento de documentos com base em intervalos de valores de *shard key* para uma determinada coleção compartilhada; definir o intervalo para cada uma das *shards zones* (as informações de cada área do conhecimento para o seu respectivo *shard*); e verificar as configurações do banco de dados.

Após a configuração do banco de dados, foi desenvolvido e aplicado o algoritmo de inserção das informações no banco de dados. Com isso, foram armazenados todos os *abstracts*, em que cada documento do MongoDB contém as informações de um resumo científico.

Com base nos dados armazenados, foi desenvolvida a aplicação para apresentar as informações, a qual lista as informações dos *abstracts* em um formato de tabela,

possibilitando o filtro pela área do conhecimento, disciplina de cada uma das áreas do conhecimento ou uma busca digitável pelos campos de área do conhecimento, disciplina, revista, título do arquivo, ano de publicação e data de publicação, além de permitir visualizar e baixar o texto do resumo científico inicial.

Com o objetivo de comprovar a utilização do banco de dados MongoDB com o método *Sharding*, foi dimensionado um conjunto de testes utilizando a aplicação desenvolvida e também o MongoDB Compass. Devido ao *Sharding* possuir a característica de direcionar uma consulta ao respectivo *shard*, caso a chave de fragmento seja utilizada, desenvolveu-se a funcionalidade de *download* e visualização detalhada das informações do registro da tabela, as quais utilizam a chave de fragmento para realizar a consulta e obter as informações referentes ao registro. Logo, para demonstrar o comportamento da consulta com a chave de fragmento, primeiro utilizou-se uma consulta baseada na área do conhecimento e no identificador do documento. A seguir efetuou-se uma consulta sem a chave de fragmento, apenas com o identificador. Para tanto, empregou-se o MongoDB Compass, já que ele permite explorar os dados do MongoDB e realizar filtros de consulta, exibindo o seu plano de execução. Com isso, permite-se a comparação das duas consultas ao executar as funcionalidades de visualização e *download*. Uma das consultas utiliza a chave de fragmento e a outra não.

Além disso, foram realizados os testes de tempo de execução da busca por disciplina com e sem a sua respectiva área do conhecimento. Para a realização desses testes, foi informada a disciplina quando selecionado a opção de listagem de todas as áreas do conhecimento e, posteriormente, informada a disciplina dentro da sua respectiva área do conhecimento. Na primeira busca, a aplicação utiliza apenas a disciplina como filtro para realizar a consulta, porém na segunda busca é utilizada a disciplina e sua respectiva área do conhecimento.

Sendo assim, os testes propostos apresentam o tempo de execução da consulta por disciplina com e sem a sua respectiva área do conhecimento, além de demonstrar o funcionamento do direcionamento da consulta para o respectivo *shard* quando realizada a busca para visualizar as informações ou realizar o *download* do arquivo.

Resultados e discussão

A coleta dos novos *abstracts* na Web of Science resultou, juntamente aos resumos coletados na primeira amostra, na mesma quantidade de arquivos entre as três áreas (*Field*) do conhecimento: Ciências Biológicas (*Biological Sciences*), Ciências Exatas (*Hard Sciences*) e Ciências Sociais e Humanas (*Social & Human Sciences*), conforme Tabela 2.

Tabela 2: Complemento do *corpus* de estudo

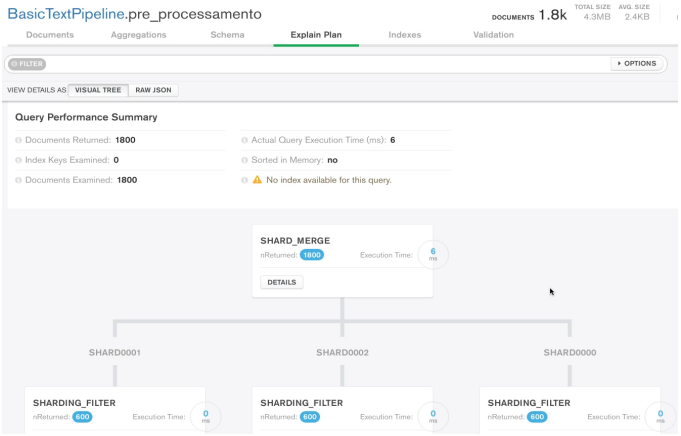
Field	***Discipline***	***Journals***	***Abstracts***
Biological Sciences	*Biology*	5	100
	Nutrition & Dietetics	5	100
	Medicine, research & experimental	5	100
	Neurosciences	5	100
	Psychology, Multidisciplinary	5	100
Social & Human Sciences	*Communication*	5	100
	Education, scientific disciplines	5	100
	Information science & library science	5	100

Fonte: Autores (2020)

A coleta dos *abstracts* realizada possibilitou igualar o número de resumos por área do conhecimento.

Além da coleta dos dados, com a elaboração da modelagem e estruturação das informações do banco de dados, foi possível obter uma divisão equivalente entre os três *shards* das áreas do conhecimento, já que foi configurado um banco de dados com a divisão da coleção de pré-processamento pela área do conhecimento. Desse modo, cada *shard* armazenou os 600 *abstracts* da sua respectiva área do conhecimento, totalizando assim 1.800 *abstracts*, conforme Figura 3.

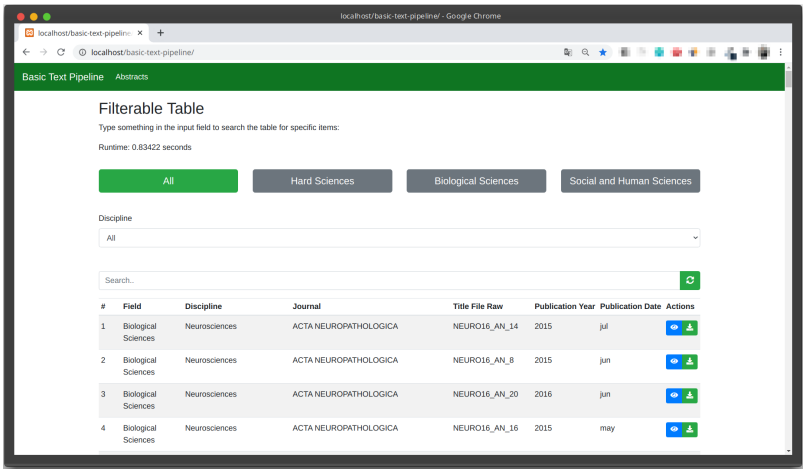
Figura 3: Visualização pelo MongoDB Compass dos dados armazenados do banco de dados nos *shards*



Fonte: Autores (2020)

O banco de dados possibilitou armazenar as informações de cada um dos *abstracts*, além de todo o conteúdo do arquivo coletado na Web of Science, o que permitiu que fosse desenvolvida uma aplicação para visualizar as informações armazenadas e também para comprovar a utilização do *Sharding*. Isso possibilita ao cientista de dados utilizar a ferramenta para a validação dos dados pré-processados e armazenados no banco de dados, podendo anotá-los e analisá-los, de acordo com a demanda de análise linguística. O uso de tecnologia auxilia e ancora o trabalho do linguista, além de permitir visualizar os dados, entender e acompanhar o trabalho realizado pelo cientista de dados. A Figura 4 demonstra a tela inicial da aplicação.

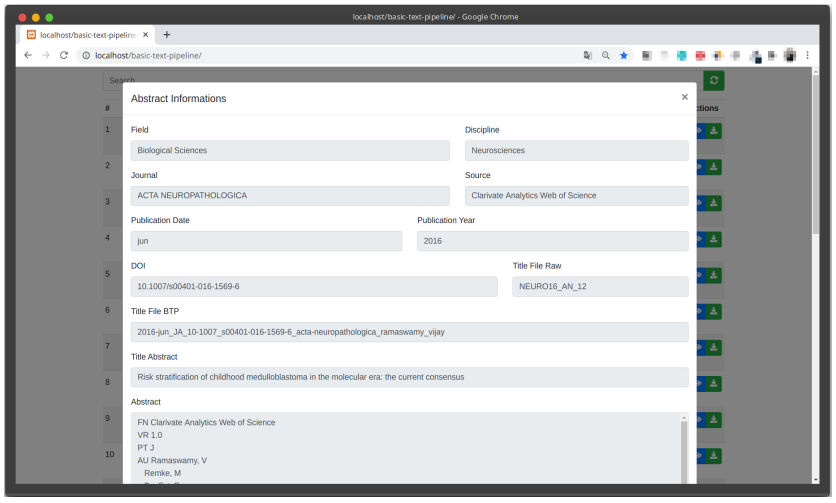
Figura 4: Tela inicial da aplicação



Fonte: Autores (2020)

Além disso, foram desenvolvidos filtros por área do conhecimento, permitindo buscar por apenas uma área do conhecimento em específico ou por todas; filtros por disciplina, permitindo a busca por uma das seis disciplinas da área do conhecimento, de acordo com a área do conhecimento selecionada. Por fim, foram desenvolvidas as funcionalidades de visualizar as informações detalhadas daquele documento, funcionalidade do *download* do arquivo contendo as informações do resumo coletado na Web of Science e a funcionalidade de filtrar dinamicamente as informações enquanto o texto é digitado. A Figura 5 mostra um exemplo da tela exibida ao clicar no botão de visualizar as informações detalhadas de um documento.

Figura 5: Tela de visualização detalhada das informações



Fonte: Autores (2020)

Na aplicação, existe a informação "*Runtime*", que é o tempo de execução quando é realizada a consulta pelo filtro da área do conhecimento ou disciplina. A informação foi inserida para demonstrar o tempo de execução quando é realizada a busca no banco de dados por uma disciplina com e sem a utilização em conjunto da área do conhecimento.

Para os testes do tempo de execução, foram realizadas as buscas por cada uma das disciplinas sem a utilização em conjunto da área do conhecimento, selecionando a disciplina quando informadas todas as áreas do conhecimento. Além disso, foram realizadas as buscas por cada uma das disciplinas com a utilização em conjunto da área do conhecimento, selecionando a disciplina quando informada a área do conhecimento a qual ela pertence.

Com base no tempo de execução de cada uma das buscas por disciplina, foi verificado que todas as buscas que utilizam a área do conhecimento em conjunto apresentaram o tempo de execução mais rápido do que as buscas sem utilizar a área do conhecimento. Entretanto, as consultas realizadas foram executadas de maneira local, já que o desenvolvimento da aplicação e a configuração do banco de dados foram realizadas todas em um único hardware. As configurações desse hardware são de 12GB de Memória RAM, Processador Core i7 de oitava geração e SSD NVMe M.2 de 240GB, além da utilização do sistema operacional Ubuntu 20.04 LTS. A Tabela 3 demonstra os tempos em segundos obtidos em cada uma das formas de busca por disciplina.

Tabela 3: Tempo de execução busca por disciplina com e sem área do conhecimento[5]

Field	***Discipline***	**Tempo sem *Field***	**Tempo com *Field***
Hard Sciences	*Agronomy*	0.56779	0.20028
	...	...	...
Social and Human Sciences	...	...	…
	Education, scientific disciplines	0.53886	0.21849

Fonte: Autores (2020)

[5] Tabela completa disponível em https://drive.google.com/file/d/109-YODEKsyMZQYJMHCa1eS9KWmVMO8kB/view?usp=sharing

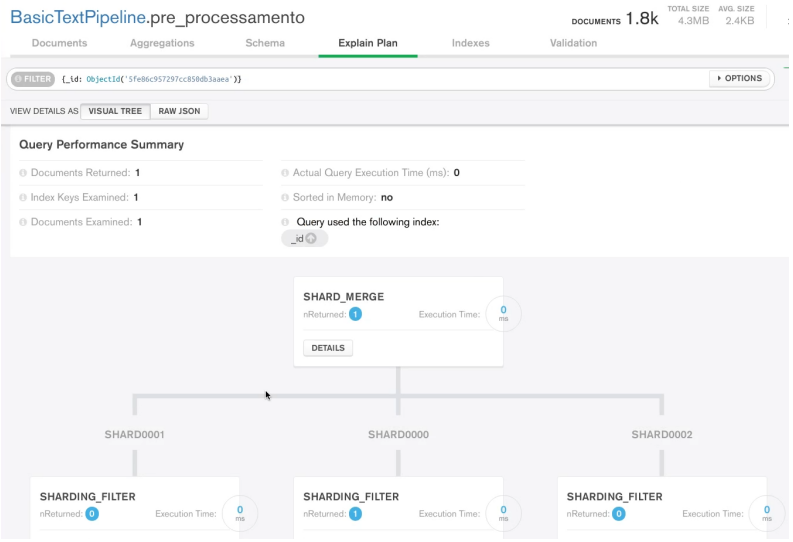
Portanto, verificou-se que, ao utilizar a área do conhecimento na busca por disciplina, todas apresentaram o tempo de execução menor do que quando não utilizada, já que o *Field* faz parte da *shard key* composto e com isso o próprio mongo consegue direcionar melhor as buscas para os *shards* que contém a respectiva área do conhecimento, tornando as buscas mais rápidas.

Além disso, o Mongos[6] conseguem direcionar a busca para o respectivo *shard* quando é utilizada a *shard key* na consulta. Com isso, foram desenvolvidas as funcionalidades de visualização detalhadas das informações e a funcionalidade de *download*, pois em ambas são utilizadas as informações da *shard key* para realização da busca, ou seja, tanto o *Field* quanto o identificador do documento.

Com base nessas buscas, foi possível verificar no MongoDB Compass, através da função "*Explain Plan*", como é realizada a consulta de maneira detalhada, ou seja, quais *shards* foram consultados para retornar as informações solicitadas na busca. Logo, foram realizados: a busca, utilizada nas funcionalidades de visualização, o *download* da aplicação, além da demonstração das consultas por um documento utilizando somente o identificador sem a área do conhecimento. A Figura 6 mostra as consultas realizadas utilizando apenas o identificador do documento para um registro da *Hard Sciences*.

Figura 6: Consulta documento *Hard Sciences* sem área do conhecimento

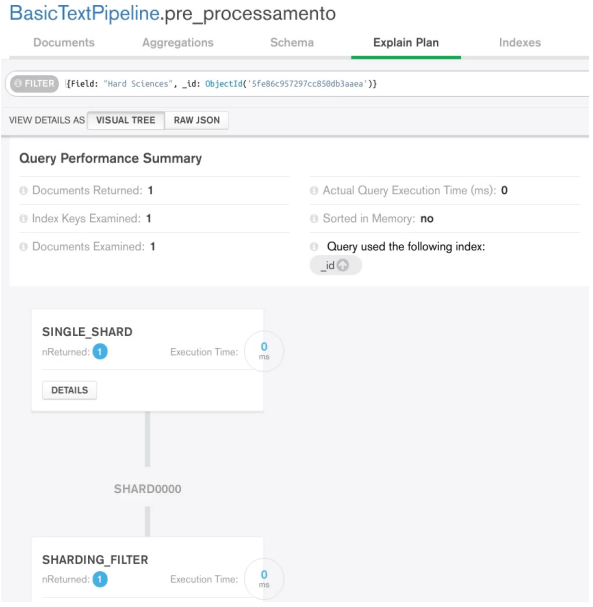
[6] O Mongos atua como um roteador de consulta, fornecendo uma interface entre os aplicativos clientes e o *cluster* fragmentado (Sharding, 2021)



Fonte: Autores (2020)

Posteriormente, foram realizadas as consultas utilizando a área do conhecimento e o identificador do documento. A Figura 7 demonstra a consulta realizada para um documento da *Hard Sciences*.

Figura 7: Consulta documento *Hard Sciences* com área do conhecimento



Fonte: Autores (2020)

A partir das duas maneiras de realização de consulta, verificou-se que em ambas foi examinado apenas um documento, como demonstrado pelo campo "*Documents Examined*", além de ser examinada apenas uma chave de índice, conforme o campo "*Index Keys Examined*". Entretanto, na segunda consulta, ao se utilizar a área do conhecimento junto ao identificador do documento, verificou-se que a consulta foi direcionada apenas para o *shard* com a respectiva informação, nesse caso o *shard* de *Hard Sciences*.

Posteriormente, foram realizadas as consultas para documentos das áreas do conhecimento de *Biological Sciences* e *Social and Human Sciences* e os resultados obtidos para ambas foram similares ao de *Hard Sciences*. Ao utilizar a área do conhecimento, juntamente com o identificador dos documentos, as consultas foram direcionadas aos respectivos *shards*.

Por fim, com os testes realizados utilizando a aplicação e o MongoDB Compass, verificou-se que o *Sharding* possibilita consultas mais rápidas quando utilizado o campo pertencente à *shard key*, além de direcionar a consulta somente ao *shard* quando é utilizada a *shard key*.

A partir da característica apresentada pelo *Sharding*, as consultas de dados apresentam melhor desempenho, já que conseguem direcionar somente ao *shard* respectivo, possibilitando assim que esse desempenho, acesso e manipulação dos dados se mantenha com o crescimento do volume de dados da Basic Text Pipeline.

O levantamento dos dados do pré-processamento permitiu o armazenamento das informações no banco de dados de maneira que os dados estivessem organizados para recuperação e identificação, possibilitando a divisão das informações por área do conhecimento. Além disso, a característica do MongoDB de armazenar os dados em documentos BSON torna possível a flexibilidade e a fácil manipulação para a inserção de novas informações, permitido não apenas o armazenamento de *abstracts*, mas também de capítulos de livros e demais textos que forem necessários, já que o MongoDB armazena os dados de maneira semi-estruturada e possui o método GridFS, que armazena arquivos independentemente do seu tamanho.

Conclusão

Esta pesquisa possibilitou a modelagem computacional para o pré-processamento da Basic Text Pipeline com MongoDB e *Sharding*, visando implementar o banco de dados e disponibilizar os dados para utilização na Basic Text Pipeline e, assim, padronizar o local de armazenamento das informações. Isso torna possível ao cientista de dados sua manipulação e validação, a fim de utilizá-los para uma aplicação linguística tecnológica para auxiliar o trabalho do linguista, que envolve aplicar os resultados dos dados de análise na produção de ferramentas tecnológicas linguísticas para o ensino de redação científica.

A limitação desta pesquisa foi que a configuração do banco de dados poderia ter sido realizada em *hardwares* diferentes para analisar o comportamento do banco em servidores diferentes, mas só foi possível realizar a configuração com esse equipamento. Pretende-se implementar a utilização do *sharding* em *hardwares* diferentes, além de ampliar a inserção dos dados das fases de processamento e também de pós-processamento da Basic Text Pipeline, já que o banco de dados é flexível para o armazenamento de dados não-estruturados.

Por fim, conclui-se que o MongoDB viabiliza o armazenamento de diferentes dados de maneira mais flexível e com menos manutenção e, além disso, juntamente ao *Sharding*, possibilita a divisão dos dados entre servidores, uma vez que se verificou que as consultas são direcionadas ao respectivo *shard* quando utilizada a chave de fragmento. Apresenta, assim, melhor desempenho de tempo de acesso e busca, já que consulta apenas os dados do respectivo *shard* e, através do índice, limita o número de documentos que deve inspecionar.

Referências

ALVES, M. das D. R.; SOUZA, M. I. F. Estudo de correspondência de elementos metadados: DUBLIN CORE e MARC 21. *RDBCI: Revista Digital De Biblioteconomia E Ciência Da Informação*, *5*(1), 20–38. https://doi.org/10.20396/rdbci.v4i2.2019. 2007.

BANKER, K. *MongoDB in Action* (1st Ed.). Manning Publications. 2011.

CATTELL, R. Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record*, 39(4). http://www.cattell.net/datastores/Datastores.pdf. 2010.

CUNHA, T. M. de A. Escalabilidade de sistemas com banco de dados nosql: um estudo de caso comparativo com mongodb e mysql. [trabalho de conclusão de curso, – Centro Universitário da Bahia – Estácio, Salvador]. Repositório de pesquisa – Centro Universitário da Bahia – Estácio, Salvador. 2011.

DATE, C. J. *Introdução a Sistemas de Bancos de Dados* (8th Ed.). Elsevier. 2004.

DOCS.MONGODB. *Get started with MongoDB*. Docs.MongoDB. https://docs.mongodb.com/. 2021.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados* (6th Ed.). Pearson Universidades. 2010.

HOWS, D.; MEMBREY, P.; PLUGGE, E. Introdução ao MongoDB (1st Ed.). Novatec. 2015.

INTEL. *Curso Big Data*. Intel. http://dialogoti.intel.com/pt-br/curso/big-data. 2015.

LÓSCIO, B. F.; OLIVEIRA, H. R. de; PONTES, J. C. de S. Nosql no desenvolvimento de aplicações web colaborativas. *Simpósio Brasileiro de Sistemas Colaborativos – SBSC*. 8 https://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf. 2011.

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de Gerenciamento de Bancos de Dados* (3rd Ed.). AMGH. 2007.

SETZER, V. M.; SILVA, F. S. C. da. *Bancos de Dados: Aprenda o que São, Melhore seu Conhecimento, Construa os Seus* (1st Ed.). Blucher. 2005.

SETZER, V. W. Dado, informação, conhecimento e competência. *DataGramaZero*, 0(0). http://hdl.handle.net/20.500.11959/brapci/7327. 1999.

SHARDING. *Sharding*. Docs.MongoDB. https://docs.mongodb.com/manual/sharding/. 2021.

SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. *Sistema de Banco de Dados* (6th Ed.). Elsevier. 2012.

VAZ, M. S. M. G. *Metamídia – um modelo de metadados na indexação e recuperação de objeto multimídia. [*tese de doutorado, Universidade Federal de Pernambuco]. Repositório de pesquisa Universidade Estadual de Ponta Grossa. http://ri.uepg.br/riuepg/bitstream/handle/123456789/638/TESE_MariaSaleteMarconGomesVaz.pdf. 2000.