Estudo da frequência de mudança de fórmulas matemáticas na *Wikipedia*

Herbert Habermann[1], **Flavio B. Gonzaga**[1]

[1]Universidade Federal de Alfenas (UNIFAL-MG)
Alfenas – MG – Brasil

{herbert.habermann,fbgonzaga}@bcc.unifal-mg.edu.br

Resumo. *Ferramentas de busca textual estão presentes em nosso cotidiano desde o surgimento e popularização da Internet. Contudo, essas ferramentas de busca textual são insuficientes no contexto de conteúdo diversos, como por exemplo imagens, fórmulas matemáticas, entre outros. Assim, começaram a surgir ferramentas de busca de conteúdos específicos. Para estas ferramentas serem efetivas, é essencial que sua base de dados esteja atualizada de acordo com o conteúdo presente na Internet. Entretanto, os estudos de frequência de atualização estão muito voltados para busca textual, se fazendo necessário um aprofundamento voltado a conteúdos específicos, como por exemplo as fórmulas matemáticas. Assim, este trabalho buscou medir com que frequência novas fórmulas são adicionadas ou removidas da Wikipedia versão inglês, com intuito de prever qual seria a taxa que a base de dados se torna desatualizada no decorrer do tempo. Para isso foi obtido o históricos de edição das páginas que possuem fórmulas matemáticas, e organizados no banco de dados em ordem cronológica. A partir disso foi possível realizar uma análise com base na data de geração dos dumps, a fim de obter as taxas de crescimento e de mudanças de fórmulas matemáticas.*

Abstract. *Textual search tools are constantly present since the Internet emergence and popularization. However, these textual search tools are insufficient for a variety of contents, such as images, mathematical formulas, among others. Thus, specific content search tools began to appear. For these tools to be effective, it is essential that their database is updated in accordance with the content present on the Internet. However update frequency studies are very focused on textual content, studies focusing on specific content are needed, such as for mathematical formulas. Thus, this work sought to measure how often new formulas are added or removed from Wikipedia in the English version, in order to predict what would be a rate at which the database becomes outdated, over the time. For this, the editing histories of the pages that have mathematical formulas were obtained, and organized in a database in chronological order. Then, it was possible to perform an analysis based on data generated from the dumps, in order to obtain the growth and change rates for mathematical formulas.*

1. Introdução

A Internet apresentou crescimento significativo nos últimos anos, sendo este um resultado da popularização da tecnologia [Chaitra et al., 2020]. O alto volume de dados da Internet tornou os mecanismos de buscas indispensáveis, pois as informações na web são dinâmicas, diversas e não estruturadas [Kolobov et al., 2019]. Nesse contexto, os buscadores desempenham um papel crucial ao baixar e indexar conteúdo da *web* com o objetivo de oferecer formas mais rápidas e precisas ao usuário no momento de acesso à informação [Bekkerman and Gilpin, 2013].

Desde o surgimento da Internet, ferramentas para busca textual vêm sendo criadas e/ou aprimoradas [Sharma and Gupta, 2015]. Os exemplos vão desde ferramentas de busca tradicionais, como o Google e Yahoo!, até propostas como a Duck Duck Go. Com o passar dos anos, no entanto, o surgimento de novos tipos de conteúdo fez com que a busca textual se tornasse insuficiente para atender às demandas dos usuários em alguns cenários específicos [Singhal et al., 2001]. Assim surgiram ferramentas de busca capazes de indexar tipos específicos de conteúdo, que vão além de somente texto [Maleki-Dizaji et al., 2014]. Alguns exemplos são o Google Imagens, onde uma consulta retorna apenas imagens, Google Scholar, cujo objetivo é a busca por artigos científicos, e o SearchOnMath, uma ferramenta de busca para texto e/ou fórmulas matemáticas, entre outras.

Apesar dos diferentes tipos de ferramentas de busca destacados no parágrafo anterior, uma característica comum a todos é a manutenção do seu índice local, que são as páginas previamente baixadas e indexadas no banco de dados, pois devem estar sempre o mais atualizado possível [Radinsky and Bennett, 2013]. Essa é uma tarefa desafiadora, uma vez que páginas diferentes possuem taxas específicas de atualização de conteúdo ao longo do tempo [Grimes et al., 2008]. Assim, desenvolver um *web crawler* que seja capaz de fazer a atualização do índice local da ferramenta de busca de maneira otimizada e precisa, constitui tema de pesquisa na área de Busca e Recuperação de Informação [Agbele et al., 2016]. O presente trabalho, no entanto, analisa um outro problema: O quão desatualizado se torna um índice que obrigatoriamente só pode ser atualizado mensalmente? Este questionamento faz sentido ao se indexar sites como *Wikipedia*, fóruns do Stack Exchange, artigos do arXiv, dentre outros; onde o uso de web crawler é desaconselhado, e a liberação de *dumps* para fins de atualização é feita mensalmente.

Na literatura existem diversos trabalhos que analisam a frequência com que *web crawlers* devem ser executados visando manter um site de busca o mais atualizado possível, além de otimizar o tráfego na busca por novas atualizações [Zineddine, 2016], porém no cenário de sites educacionais esses estudos são mais escassos. Tendo isso em vista, o presente trabalho realiza a análise no contexto de busca matemática, onde buscadores como a SearchOnMath[1], Approach0[2], acabam tendo que atualizar parte do seu índice com frequência miníma mensal, uma vez que dependem da liberação de *dumps*.

Assim, o objetivo deste trabalho é medir com que frequência novas fórmulas matemáticas são acrescentadas ou removidas da *Wikipedia* versão em inglês, considerando a frequência com que *dumps* são disponibilizados na mesma. Tais números fornecem uma aproximação do quão defasadas as bases de dados de tais buscadores se tornam, caso

[1] `https://www.searchonmath.com`
[2] `https://approach0.xyz/search/`

os mesmos sejam atualizados mensalmente (intervalo mínimo possível) ou em intervalos de tempos ainda maiores (bimestralmente, trimestralmente, e assim por diante). Por fim, o trabalho ainda apresenta um estudo sobre a evolução do conteúdo matemático na *Wikipedia* desde o seu início até os dias atuais.

2. Trabalhos Relacionados

Na literatura existem trabalhos cujo foco principal é a determinação de taxas de atualização de bases locais para buscadores textuais. O foco desses estudos é estabelecer qual o melhor intervalo de tempo para que ferramentas de busca atualizem seus dados locais, para que esses não fiquem defasados. O grande desafio está na diversidade de conteúdo presentes na *web* e a constância com que eles são atualizados, por exemplo, sites de notícia, são atualizados quase que a cada minuto, enquanto sites de conteúdos educacionais são alterados em intervalos de tempo maior.

Sethi [2021] observou que a utilização de *web crawlers* em larga escala demanda de uma grande largura de banda, e que a maioria dos trabalhos focados em reduzir o consumo de Internet de *web crawlers*, sem impacto significativo nos resultados, obtinham resultados satisfatórios apenas aplicado em pequena quantidade de páginas. Foi então proposta uma alteração na estrutura de *web crawler* convencional, adicionando dois principais componentes, agendador de visitas e priorizador de urls.

De acordo com Pavai and Geetha [2017], os métodos existentes de *web scraping* atualmente são incapazes de manter as bases de dados locais atualizadas. A partir disso, foi proposta uma abordagem probabilística baseada em um *crawler* incremental, onde são monitoradas as páginas que já foram rastreadas uma vez. Para lidar com as mudanças da *web*, foi adicionada uma nova variável chamada de 'Crawl hit rate' visando avaliar a eficiência do *crawler*. De maneira geral, essa variável sinaliza se o download da página no tempo estimado, era realmente necessário,ou seja, se houve alterações na página ao executar o rastreamento. A partir desse ponto, um conjunto ponderado determina se é possível aumentar ou diminuir o tempo previsto a fim de reduzir o uso de rede total sem comprometer a atualização das informações. Quando comparado a *web crawlers* incrementais sem modificação, o banco de dados estava tão atualizado quanto, porém com uma diminuição de aproximadamente 23% no tráfego de rede.

A maioria dos trabalhos pressupõe que é necessário o conhecimento das taxas de atualização das páginas previamente. De maneira contrária, Avrachenkov, Patil, and Thoppe [2020] propõe que todas as páginas sejam atualizadas independente dessas taxas. Assim, foram propostas três abordagens, a primeira baseada na Lei dos Grandes Números (LLN), a segunda nos princípios da Aproximação Estocástica (SA), enquanto a terceira é uma extensão da segunda com um termo de momento adicional (SAM). Os experimentos foram baseados em conjuntos de dados reais do histórico de edições da *Wikipedia*.

Ao final dos estudos, Avrachenkov, Patil, and Thoppe [2020] concluíram que seus algoritmos construídos possuíam uma estimativa computacionalmente eficiente quando comparado ao MLE (Maximum Likelihood Estimator) proposto por Cho and Garcia-Molina [2003], além disso cada um dos algoritmos se encaixava melhor em diferentes contextos. Como por exemplo em crawler de alta frequência todos eles demonstraram resultados satisfatórios, já em baixa frequência o algoritmo baseado na LLN foi quem demonstrou melhores resultados.

Na literatura também são encontradas técnicas baseadas na longevidade da informação, a fim de prever sua taxa de atualização. Olston and Pandey [2008] dividiram as páginas em dois tipos de conteúdo, as de conteúdo efêmero e as de conteúdo persistente. O conteúdo efêmero não é vantajoso ser rastreado, porque muda constantemente, assim no momento em que são indexados, não são mais representativos da página da *web* da qual foi adquirido. Já o conteúdo persistente se mantém por várias atualizações da páginas por um período prolongado de tempo.

A partir disso foram desenvolvidas novas políticas de atualização de base de dados levando em consideração a longevidade da informação das páginas a fim de obter um menor custo computacional. Foi então construído um algoritmo adotando o método de shingling [Broder et al., 1997] para fragmentar cada página e calcular a divergência de informação dos fragmentos [Olston and Pandey, 2008].

Assim, foi possível otimizar a questão da divergência de informação, visto que as páginas com informações relevantes mais longínquas teriam menos prioridade de atualização em relação ao tempo gasto, priorizando apenas páginas com informações mais relevantes [Olston and Pandey, 2008].

3. Metodologia

3.1. Obtenção, extração e organização dos dados

Para a realização do trabalho proposto, três tarefas principais precisaram ser realizadas:

1. Obtenção e importação de uma versão atual da *Wikipedia* (versão em inglês)
2. Identificação das páginas que possuem fórmulas matemáticas
3. Construção do histórico destas páginas

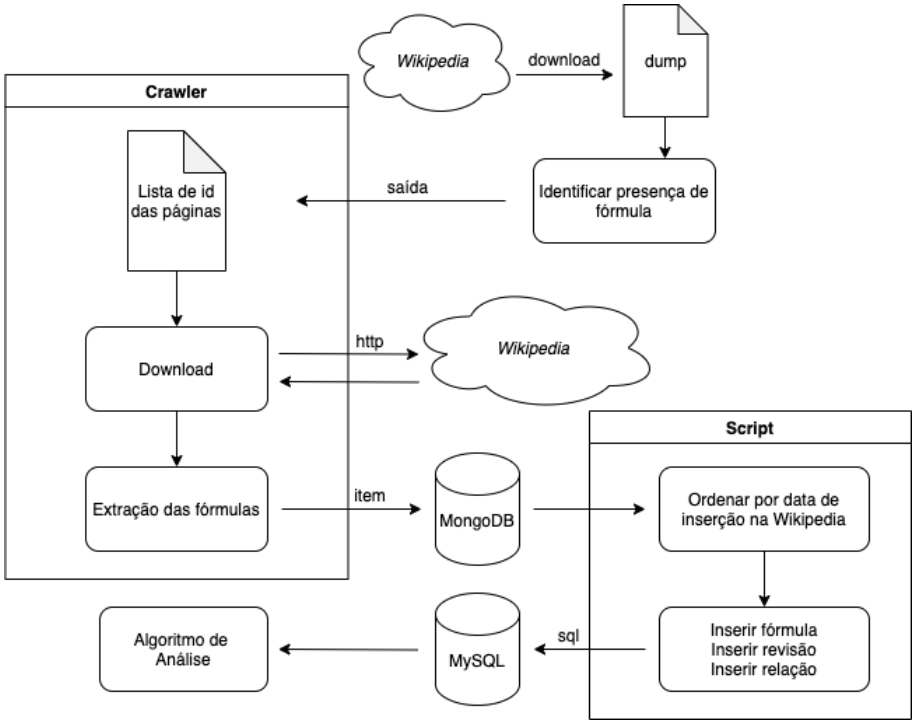


Figura 1. Fluxo de importação e organização das fórmulas matemáticas

A Figura 1 exibe uma síntese dos passos que serão descritos nesta seção. A *Wikipedia* disponibiliza, com periodicidade mensal, arquivos de *dump* contendo a versão mais atual das páginas do site. Para a realização do trabalho, foi obtido o *dump* de abril de 2021. É importante ressaltar que as páginas que no passado já tiveram fórmulas, mas que no momento da geração do *dump* analisado, não tinha nenhuma fórmula, não foram analisadas. Uma vez feito o download, o mesmo foi importado para uma base de dados MySQL, mantendo a estrutura original. A estrutura utilizada pela *Wikipedia* é composta por três tabelas principais:

- page: possui informações importantes de uma página, como por exemplo: título, data de criação, identificador único (id), entre outras.
- revision: contém os metadados das edições realizadas numa determinada página.
- text: contém os textos da página em si.

Para a identificação das fórmulas matemáticas, desenvolveu-se um algoritmo em Java que verifica em cada uma das páginas pela existência (ou não) da *tag* $$$$, que é o delimitador de fórmulas usado pela *Wikipedia*. Sempre que a *tag* era encontrada em uma página, o id da mesma era armazenado, de modo que ao término da execução do algoritmo, obtém-se os ids de todas as páginas que possuem ao menos uma fórmula matemática. Esta lista de ids foi então utilizada na próxima etapa do processo, que consistiu na construção do histórico destas páginas.

Para realizar a obtenção do histórico de edições das páginas, foi desenvolvido um algoritmo na linguagem Python utilizando o framework Scrapy, onde a partir da lista de id das páginas obtidas na etapa anterior, foram realizadas requisições HTTP na API principal da *Wikipedia*. Para realizar as requisições na API da *Wikipedia*, os seguintes parâmetros são necessários na URL:

- action: identifica qual ação deve ser realizada com a requisição.
- prop: especifica qual o agrupamento de dados.
- rvprop: especifica quais campos devem conter no agrupamento.
- format: especifica o formato de devolução dos dados.
- pageids: identificador único da página.

Como o objetivo era a obtenção dos textos de cada revisão das páginas que possuem ao menos uma fórmulas matemática atualmente, os seguintes valores foram definidos para os parâmetros:

- action: query (busque dados...)
- prop: revisions (agrupe por revisões...)
- rvprop: ids|timestamp|content (retorne estes campos...)
- format: json (neste formado...)
- pageids: ${id_pagina} (para a página com este id.)

Apenas a título de exemplo, uma consulta na API que deseja obter os dados acima especificados para a página Universal Chord Theorem[3], o qual possui o id 52167339 ficará do seguinte modo: `https://en.wikipedia.org/w/api.php?action=query&prop=revisions&rvprop=ids|timestamp|content&format=json&pageids=52167339`.

[3] `https://en.wikipedia.org/wiki/Universal_chord_theorem`

Para não sobrecarregar a API e visando manter o bom fluxo de dados, foi estipulado que fossem realizadas até 5 requisições em paralelo, com intervalo aleatório entre as requisições variando de 0,1 segundo a 0,9 segundos. Para cada requisição, a API responde um corpo no formato solicitado com informações (metadados e texto) das últimas 50 edições realizadas naquela página. Dentre os metadados existe um campo que é uma chave para que se possa acessar as próximas 50 edições, caso exista. Dessa forma, o algoritmo foi implementado de modo recursivo, onde continuava a fazer requisições enquanto houver essa chave nos metadados, fazendo com que todo o histórico de revisões fosse recuperado para todas as páginas com fórmulas matemáticas.

Para cada página, o algoritmo analisou todas as revisões existentes, gerando, para cada revisão, um objeto como descrito na Figura 2. O objeto contém apenas informações como o identificador da página, identificador da revisão, data da revisão e as fórmulas presentes naquela revisão.

```
{
  "page_id": 612,
  "rev_id": 233613,
  "date": "2001-12-17",
  "formulas": [
    "p(x)=k(x-x_1)(x-x_2)",
    "x^n-a"
  ]
}
```

Figura 2. Modelo do objeto salvo no banco de dados não relacional.

Todos os objetos construídos neste processo foram salvos em um banco de dados não relacional (MongoDB), em formato de documento e na ordem que foram baixados. Neste momento foi utilizado esse modelo de banco por conta da sua performance e também a facilidade em guardar objetos (documentos). É importante salientar que as fórmulas são mantidas exatamente como estavam dentro das tags, não sendo realizado nenhum tratamento de normalização.

Concluída a obtenção do histórico das páginas, foi realizada uma etapa cujo objetivo foi de organizar, para cada página, os históricos de maneira cronológica com intuito facilitar análise. Isso porque as revisões foram obtidas de maneira paralela sem ordem definida. Ter a base de dados organizada em ordem cronológica e em banco de dados relacional torna a análise dos dados mais simples.

Para organização dos dados, foi utilizado banco de dados relacional MySQL e foram criadas quatro tabelas, semelhante ao esquema utilizado pela *Wikipedia*. A tabela page foi utilizada apenas para evitar replicação textual do título da página no banco de dados, já a tabela revision foi responsável por referenciar a qual página aquela revisão pertence, e também se relacionar com a revisão anterior, além de conter a data na qual ela foi publicada.

Foi também criada a tabela equation com os objetivos de guardar a fórmula em si e um id. Para representar quais as fórmulas que estavam contidas em cada uma das revisões, foi criado a tabela revision_equation, contendo o id da revisão, o id da fórmula,

e a quantidade de vezes que ela aparecia naquela revisão. O modelo relacional construído é apresentado na Figura 3.

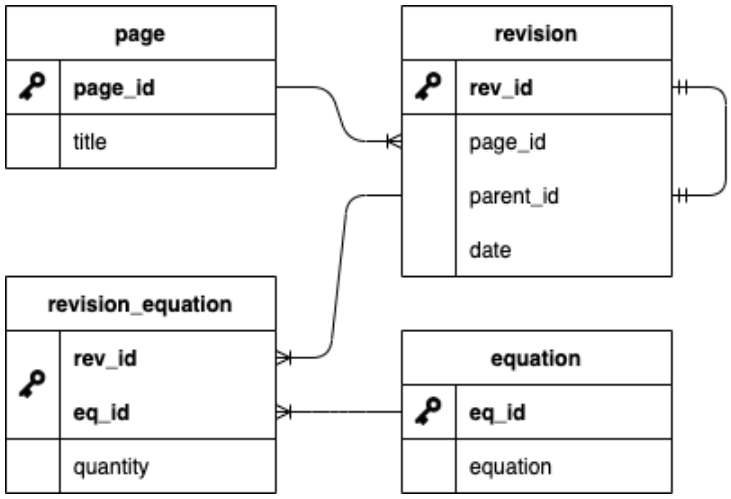


Figura 3. Esquema utilizado do banco de dados relacional

Para a migração dos dados, foi implementado um algoritmo em Python que carregou todos os objetos que estavam presentes no MongoDB, organizou-os em ordem temporal crescente, e inseriu os dados extraídos no banco de dados relacional MySQL, já indexando todas as fórmulas encontradas. Como as fórmulas foram inseridas em ordem cronológica, seu id na tabela equation, também representa a sequência em que as fórmulas surgiam na *Wikipedia*, por exemplo, a fórmula de id 100, foi a centésima fórmula inserida na biblioteca online, dentre as páginas que continham fórmulas ativas na data do *dump* analisado.

3.2. Análise dos dados

Após importação dos dados foram realizadas análises com o objetivo de verificar surgimento de formulas matemáticas na *Wikipedia* ao longo dos anos. O estudo identificou métricas importantes como a inserção e deleção de fórmulas matemáticas mês a mês desde o início suporte às tags $<math>$ em janeiro de 2003. Importante salientar que as fórmulas foram identificadas uma única vez, independente de em quantas páginas uma determinada fórmula apareça. Outro detalhe já citado, mas que é importante frisar, é que nenhum tratamento foi feito nas fórmulas. Assim, qualquer diferença na escrita originará uma nova fórmula na contagem, como por exemplo x_y será diferente de x_ y, sendo portanto contabilizadas duas vezes.

Foi construído um algoritmo para percorrer todas as datas em que são realizados os *dumps* da *Wikipedia* (dia 20 de cada mês) e registrar quais fórmulas estavam ativas naquele dia, bem como quantas foram inseridas ou deletadas de um mês para o outro.

Primeiramente, foi construída uma lista com todas as datas, desde o surgimento da primeira fórmula. Considerou-se como o surgimento da primeira fórmula a data em que a *Wikipedia* passou a dar suporte às tags $<math>$. As comparações foram feitas tomando como base sempre dois meses consecutivos.

Foram feitas iterações com cada uma das datas da lista inicial, recuperando-se o maior id de fórmulas matemática que se tem até aquele dia. Esse é um detalhe relevante,

pois os ids das fórmulas representam a ordem cronológica em que elas apareceram na *Wikipedia*. Assim, mesmo que uma fórmula seja inserida em um determinado tempo, e excluída em um tempo futuro, o seu registro permanece no banco, indicando que determinada fórmula já esteve, em algum momento, presente na *Wikipedia*.

Sendo a tabela de fórmulas construída ao longo dos meses, e respeitando-se a ordem cronológica dos ids, é possível descobrir quais fórmulas foram inseridas e quais foram excluídas tomando-se como base dois meses distintos.

Por último, foi feita a análise de fórmulas denominadas fantasmas, que são fórmulas que foram adicionadas e deletadas no intervalo entre as gerações de *dump* da *Wikipedia*. Tais fórmulas, apesar de terem existido em algum momento na *Wikipedia*, não são identificadas por algoritmos de indexação por não estarem presentes nas páginas no momento em que são gerados os *dumps*. Está análise só foi possível em virtude de terem sido obtidas todas as revisões para todas as páginas que possuíam alguma fórmula. Então, se considerar uma determinada página onde uma revisão r_1 insira uma fórmula, e uma revisão r_2 exclua essa fórmula, caso estas revisões aconteçam no intervalo entre a geração de dois *dumps* consecutivos, elas terão existido na *Wikipedia*, mas não terão sido incluídas em nenhum dos *dumps*.

4. Resultados

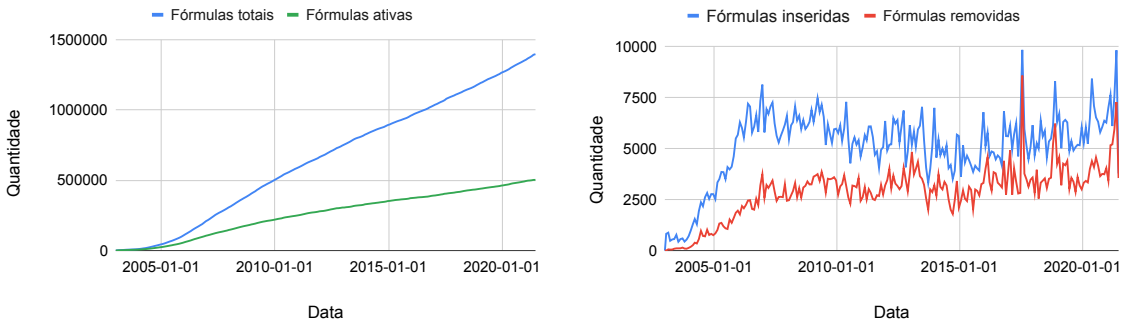
Após a execução do algoritmo de análise foi possível construir alguns gráficos para entender como é o comportamento das fórmulas matemáticas dentro da biblioteca da *Wikipedia*.

Desde o momento em que a *Wikipedia* passou a suportar fórmulas matemáticas, muitas fórmulas já foram inseridas na biblioteca. Porém, nem todas elas permaneceram ativas (disponíveis online) ao longo dos anos. A Figura 4 ilustra uma análise ao longo de todo o histórico de fórmulas dentro da biblioteca.

A Figura 4 (a) apresenta a evolução das fórmulas dentro da biblioteca, tanto com relação ao total de fórmulas (inseridas em algum momento) quanto as fórmulas que estavam ativas. A partir desses dados foi possível calcular a taxa média de crescimento que atualmente está em aproximadamente $0,54\%$. Já a figura 4 (b) apresenta o total de inserções e remoções que aconteceram mês a mês. É possível observar que nos três primeiros anos a quantidade de fórmulas inseridas apresentou um crescimento mais agressivo, e acabou se estabilizando nos anos seguintes. Em relação às remoções, existe uma tendência em acompanhar os picos de inserção. Assim, é possível inferir que na maioria dos casos onde ocorre uma inserção e deleção combinadas, possivelmente representa edição da fórmula.

Com base nos gráficos apresentados nas Figura 4, observa-se uma tendência de estabilidade tanto nas curvas de inserção, quanto na de deleção. Assim, optou-se por fazer uma análise mais detalhada do histórico recente da *Wikipedia*, onde foram considerados os últimos 5 anos.

O padrão de inserção e remoção de fórmulas matemáticas em relação a quantidade de fórmulas ativas mês a mês está apresentado na Figura 5 (a). Pode ser observado que novas fórmulas são acrescidas com uma proporção de 1.32% em comparação ao total de fórmulas no mês. Porém, são removidas com uma proporção de 0.78%. Além desses parâmetros, é importante conhecer a taxa de modificações, sendo essa obtida pela soma

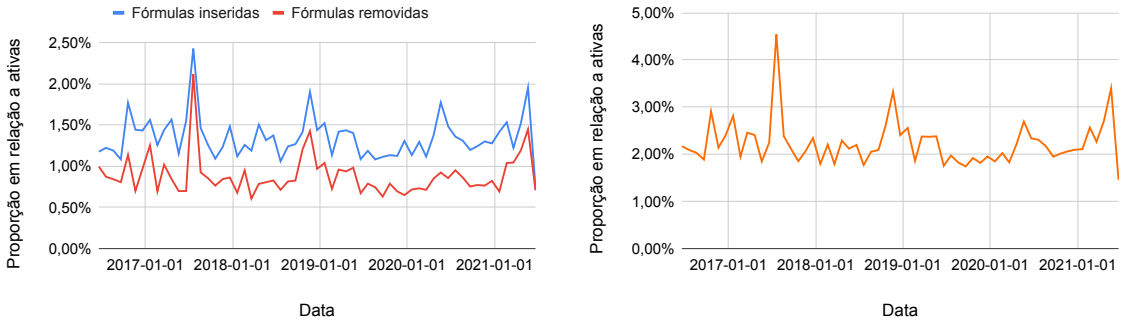


(a) Evolução

(b) Inserções e remoções.

Figura 4. Fórmulas matemáticas na *Wikipedia* ao longo dos anos.

das inserções e remoções Figura 5 (b). A partir dos dados apresentados, é possível calcular que a proporção média de modificações é de aproximadamente 2.10%.



(a) Inserções e remoções.

(b) Modificações.

Figura 5. Proporção com relação com o total de fórmulas ativas

Em termos práticos é importante prever o quão desatualizada uma base fica no decorrer do tempo. Desatualização representa a diferença entre o conteúdo indexado na base de dados local em relação ao conteúdo que está online naquele momento. Com os dados apresentados, foi observado que a quantidade de formulas ativas está em constante crescimento. Foi proposto um modelo matemático levando em consideração a taxa média de crescimento de fórmulas na *Wikipedia* e também a taxa média de modificação conforme a equação (1):

$$Td = \sum_{t=1}^{N} (1 + Tc)^{t} Tm \tag{1}$$

Onde:

- Td: Taxa de desatualização
- Tc: Taxa de crescimento
- Tm: Taxa de modificação
- t: Tempo em meses

Assim, quando essa fórmula é aplicada é obtido a taxa de desatualização da base dados em determinado tempo. Podemos observar a aplicação da fórmula para períodos de 3, 6, 9 e 12 meses e taxa aproximada de desatualização na Tabela 1.

N	Td
3	6.37%
6	12.84%
9	19.42%
12	26.10%

Tabela 1. Taxa de desatualização da base de fórmulas matemáticas

Outro ponto observado são as fórmulas aqui denominadas de fantasmas, sua ocorrência é devido ao intervalo de *download* da *Wikipedia* ser fixo, existem fórmulas que são inseridas e removidas antes mesmo da geração de um *dump*. A Figura 6 mostra a proporção de fórmulas fantasmas em relação a quantidade de ativas no decorrer do tempo.

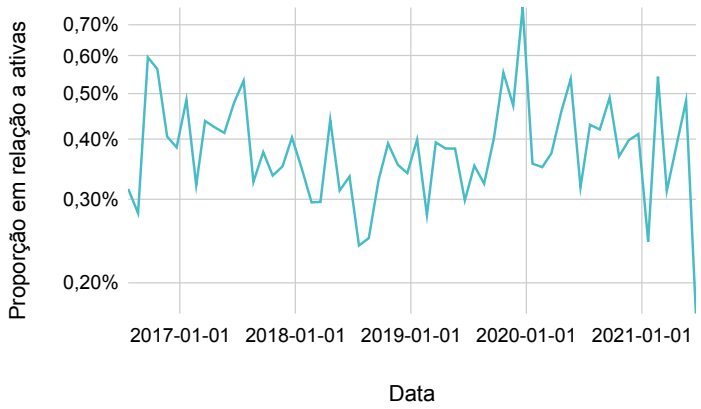


Figura 6. Fórmulas fantasmas em relação ao total de fórmulas ativas.

5. Conclusão

É possível observar que o conteúdo matemático na biblioteca *Wikipedia* passou a ter maior adesão dos colaboradores a partir de 2006 e desde então, a quantidade de fórmulas matemáticas está sempre em constante evolução. A partir da análise dos dados, foi possível observar o quão volátil as fórmulas matemáticas são na biblioteca *Wikipedia* pois já foram excluídas mais fórmulas do que a quantidade de ativas atualmente.

Nos últimos cinco anos, a frequência de inserção e remoção das fórmulas é praticamente constante com relação ao total de fórmulas ativas. Isso é um aspecto positivo para os buscadores, permitindo planejamento e definição de taxa desatualização do banco de dados. Dessa forma, foi proposto um modelo matemático para prever o quão desatualizado o banco de dados estaria no decorrer do tempo. Na literatura não foram encontrados relatos de quais seriam as taxas de desatualização consideradas aceitáveis, permitindo que os próprios buscadores de fórmulas matemáticas na *Wikipedia* estabeleçam tais taxas de desatualização, permitindo assim uma otimização na indexação do conteúdo.

De forma complementar, estudos da mesma natureza podem ser aplicados em fóruns de conteúdo matemático, cujo *dump* é disponibilizado também com frequência mensal. Nesse tipo de site, espera-se um crescimento ainda maior de fórmulas mensalmente, dada a sua natureza mais dinâmica. Como exemplos pode-se citar: MathOverflow[4], StackExchange Mathemathics[5], StackExchange Physics[6], entre outros.

Referências

Kehinde K Agbele, Eniafe F Ayetiran, Kehinde D Aruleba, and Daniel O Ekong. Algorithm for information retrieval optimization. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–8. IEEE, 2016.

Konstantin Avrachenkov, Kishor Patil, and Gugan Thoppe. Online algorithms for estimating change rates of web pages. *arXiv preprint arXiv:2009.08142*, 2020.

Anton Bekkerman and Gregory Gilpin. High-speed internet growth and the demand for locally accessible information content. *Journal of Urban Economics*, 77:1–10, 2013.

Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer networks and ISDN systems*, 29(8-13):1157–1166, 1997.

PG Chaitra, V Deepthi, KP Vidyashree, and S Rajini. A study on different types of web crawlers. In *Intelligent communication, control and devices*, pages 781–789. Springer, 2020.

Junghoo Cho and Hector Garcia-Molina. Estimating frequency of change. *ACM Transactions on Internet Technology (TOIT)*, 3(3):256–290, 2003.

Carrie Grimes, Daniel Ford, and Eric Tassone. Keeping a search engine fresh: Risk and optimality in estimating refresh rates for web pages. *Proc. INTERFACE (2008)*, 2008.

Andrey Kolobov, Yuval Peres, Eyal Lubetzky, and Eric Horvitz. Optimal freshness crawl under politeness constraints. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 495–504, 2019.

Saeedeh Maleki-Dizaji, Jawed Siddiqi, Yasaman Soltan-Zadeh, and Fazilatur Rahman. Adaptive information retrieval system via modelling user behaviour. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):105–110, 2014.

Christopher Olston and Sandeep Pandey. Recrawl scheduling based on information longevity. In *Proceedings of the 17th international conference on World Wide Web*, pages 437–446, 2008.

G Pavai and TV Geetha. Improving the freshness of the search engines by a probabilistic approach based incremental crawler. *Information Systems Frontiers*, 19(5):1013–1028, 2017.

Kira Radinsky and Paul N Bennett. Predicting content change on the web. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 415–424, 2013.

Shilpa Sethi. An optimized crawling technique for maintaining fresh repositories. *Multimedia Tools and Applications*, 80(7):11049–11077, 2021.

[4] `https://mathoverflow.net/`

[5] `https://math.stackexchange.com/`

[6] `https://physics.stackexchange.com/`

Shruti Sharma and Parul Gupta. The anatomy of web crawlers. In *International Conference on Computing, Communication & Automation*, pages 849–853. IEEE, 2015.

Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

Mhamed Zineddine. Search engines crawling process optimization: a webserver approach. *Internet Research*, 2016.