TM-MR – Plataforma para Mineração de Textos em Prontuários Médicos utilizando Java

Ranna Raabe Fernandes da Costa[1]

[1]Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brasil

ranna.raabe@gmail.com

Resumo. *TM-MR é um sistema que foi desenvolvido utilizando JavaFX, e é responsável por minerar textos de prontuários médicos digitais, com o intuito de descobrir a similaridade dos prontuários e encontrar seu padrão de escrita. Foi implementado usando algoritmos de similaridade entre Strings e técnicas de mineração de texto.*

1. Introdução

Habitualmente, em **consultas médicas**, os pacientes se comunicam com um médico especialista ou clínico geral, no intuito de descrever seus sintomas, estado físico e/ou emocional, enquanto o médico registra no **prontuário médico** do paciente os sintomas e diagnóstico de sua doença. O prontuário auxilia o médico e sua equipe de saúde para que, em futuras consultas, possam obter um histórico de sintomas e doenças que o paciente dispôs.

Neste contexto, foi pesquisado e concluído que todos os textos possuem um certo "padrão"de escrita. Pensando nisso foi desenvolvida uma ferramenta com a função de minerar prontuários médicos, a fim de comparar a similaridade dos prontuários, e encontrar, exatamente, este padrão de escrita. Esta ferramenta foi denominada de *Text Mining of Medical Record* - TM-MR.

O sistema foi desenvolvido em Java, utilizando conceitos e paradigmas da linguagem. Sua interface foi implementada utilizando JavaFX, uma tecnologia para desenvolvimento de aplicações desktop. As interfaces gráficas com o usuário foram desenvolvidas utilizando o SceneBuilder. O sistema foi implementado e testado utilizando prontuários de *Anamnese*[1] da Maternidade Escola Januário Cicco, Natal - RN, na especialidade Obstetrícia.

O objetivo principal do projeto é descobrir o quão similar são as descrições dos prontuários. Os objetivos específicos do sistema são:

- utilizar técnicas de mineração de dados para processar os dados dos prontuários;
- implementar e aplicar algoritmos de similaridade entre Strings;
- gerar gráficos com as similaridades;

2. Abordagem

O fluxo de funcionamento da plataforma foi dividido em dois passos: pré-processamento dos dados e processamentos dos dados.

[1]histórico que vai desde os sintomas iniciais até o momento da observação clínica, realizado com base nas lembranças do paciente

No **pré-processamento dos dados**, é realizada a leitura dos dados, a conversão dos arquivos, e a mineração dos textos. Logo após, é feito o **processamento dos dados** através da aplicação dos algoritmos de similaridade entre Strings.

2.1. Leitura e Conversão dos arquivos

O sistema recebe uma quantidade de arquivos de prontuários. Os prontuários enviados precisam estar no formato **.pdf**, para que o sistema funcione de forma correta. Após a submissão dos arquivos, o sistema os converte para **.txt**. Então, o arquivo está pronto para ser usado na próxima etapa.

2.2. Mineração de Texto

A mineração de texto é uma técnica utilizada para retirar do texto todas as informações desnecessárias para o processamento de dados. As informações de um texto dispõe de um "padrão" de escrita, dessa forma, a técnica de mineração de texto busca encontrar as palavras importantes seguindo esses padrões textuais.

O sistema MT-MR seguiu alguns passos na mineração de texto, que foram, na seguinte ordem:

- remoção de palavras com menos de 3 caracteres;
- remoção de caracteres *upper case*;
- remoção de acentos das palavras;
- remoção de caracteres especiais;
- remoção de palavras duplicadas;
- ordenação das palavras em ordem alfabética;

Após isto, o sistema têm os prontuários prontos para terminar o processamento. É importante destacar que, a mineração de texto é feita apenas com a seção de Anamnese dos prontuários, evitando que os dados que **não** são de interesse do sistema sejam incluídos na etapa de processamento e alterem os resultados esperados.

2.3. Algoritmos de Similaridade entre Strings

Para comparar a similaridade dos textos, foram implementados 4 algoritmos: *Cosine, Trigram, Levenshtein, Jaro-Winkler*; todos os algoritmos calculam similaridade entre Strings de uma forma particular. Sendo essas formas, então:

- **algoritmo Cosine (Singhal, A., 2001; Sidorov, G. et al., 2014; Perone, C. S., 2019):** é uma métrica usada para medir a similaridade entre textos. Matematicamente, mede o cosseno do ângulo entre dois vetores projetados em um espaço multidimensional, ou seja, quanto menor o ângulo, maior a similaridade do cosseno;
- **algoritmo Trigram (Dunning, T., 1994):** Na linguística computacional e probabilidade, um n-grama é uma sequência contígua de n itens de texto ou fala. O algoritmo de Trigram é uma sequência contígua de n (três, neste caso) itens de uma amostra;
- **distância Levenshtein (Navarro, G., 2001; Wagner, Robert A. and Fischer, Michael J., 1974):** A distância Levenshtein ou distância de edição entre duas *Strings* (duas sequências de caracteres) é dada pelo número mínimo de operações necessárias para transformar uma *String* na outra. Quando se fala "operações", considera-se inserção, deleção ou substituição de um caractere;

- **algoritmo Jaro-Winkler (Jaro, M. A., 1995; Winkler, W. E., 1990):** A métrica de Jaro-Winkler é uma medida de similaridade entre duas Strings, sendo uma variação da métrica Jaro distance;

No sistema, um (ou vários) dos algoritmos acima é escolhido para calcular a similaridade e retorna uma matriz de confusão com os valores de similaridade dos prontuários. Por fim, os resultados podem ser exportados, ou seja, a *Anamnese* dos prontuários que foram selecionados e o resultado da comparação entre cada um é salvo em um novo arquivo, como mostra na seção 3.

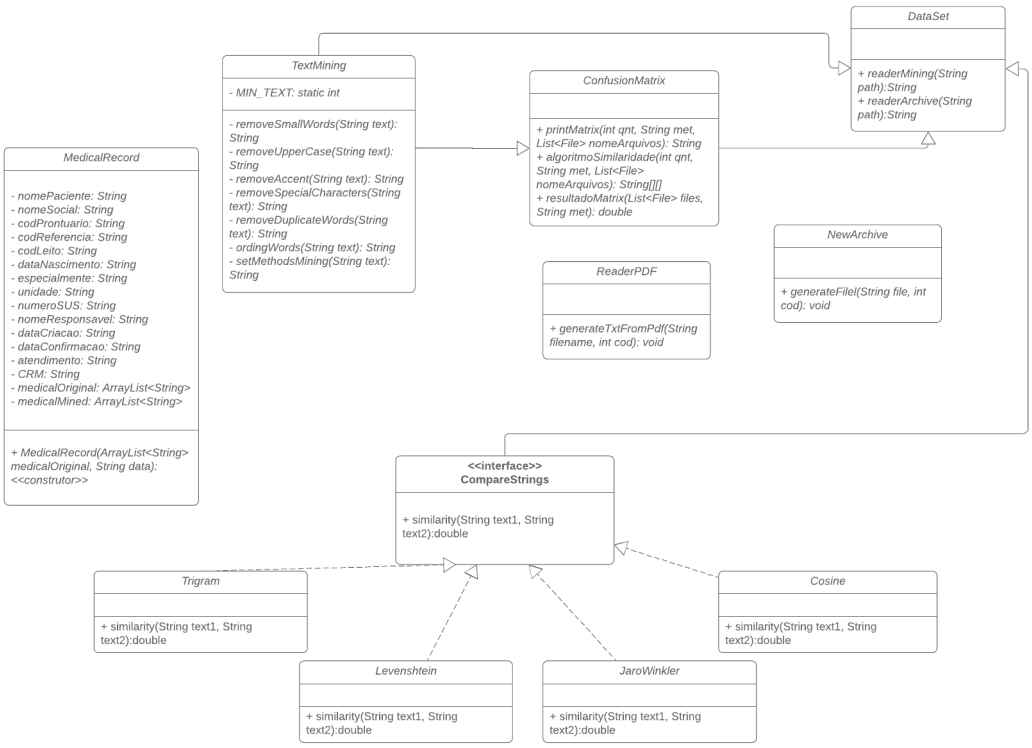
3. Descrição Geral

Este tópico descreve o funcionamento geral do sistema, divido em diagrama de classes, descrição detalhada do sistema e as experiências durante o processo de implementação.

3.1. Diagrama de classes

O diagrama de classes da plataforma está descrito na Figura 1 abaixo:

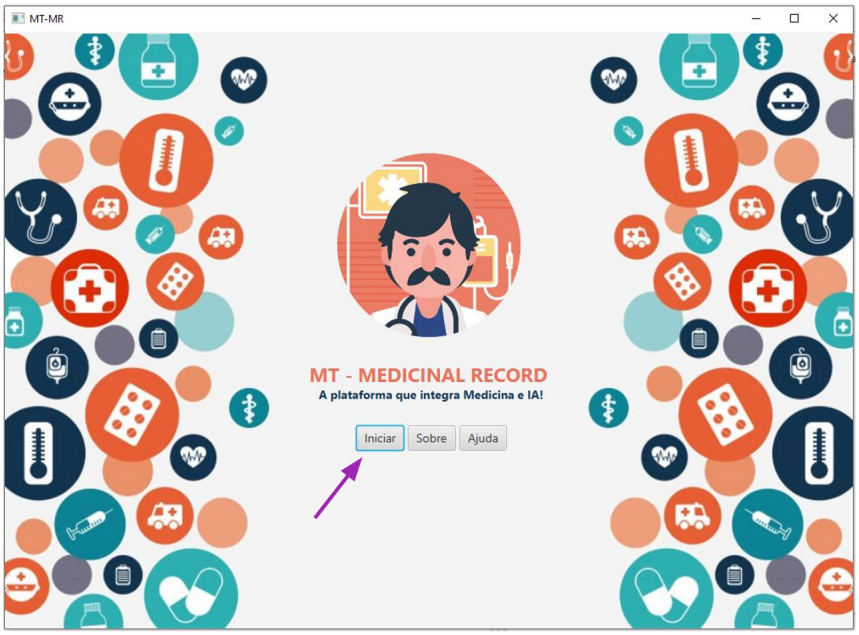
Figura 1: Diagrama de classes



3.2. Descrição do sistema

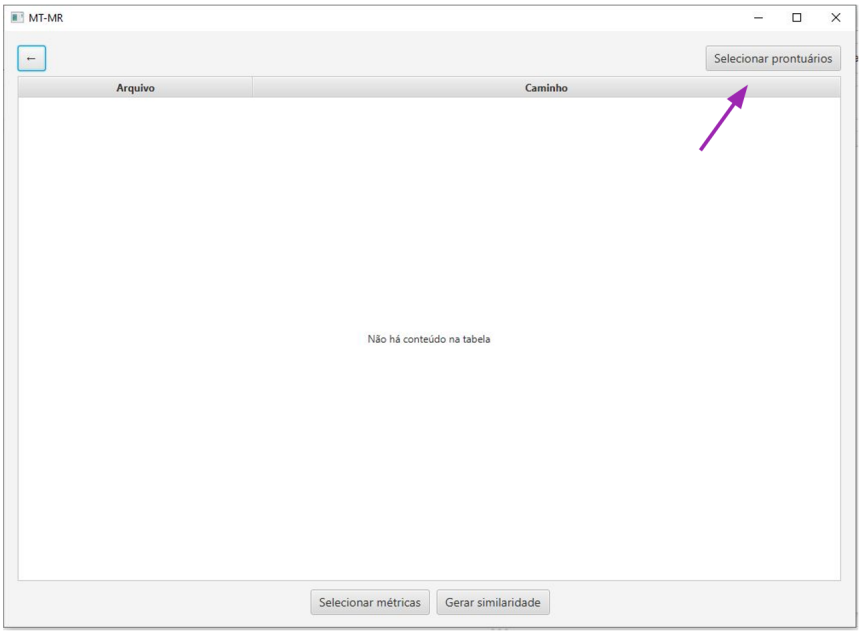
Ao executar o sistema, o usuário possui três opções na Tela Inicial: 'Iniciar', 'Sobre' e 'Ajuda' (Figura 2.

Figura 2: Tela Inicial



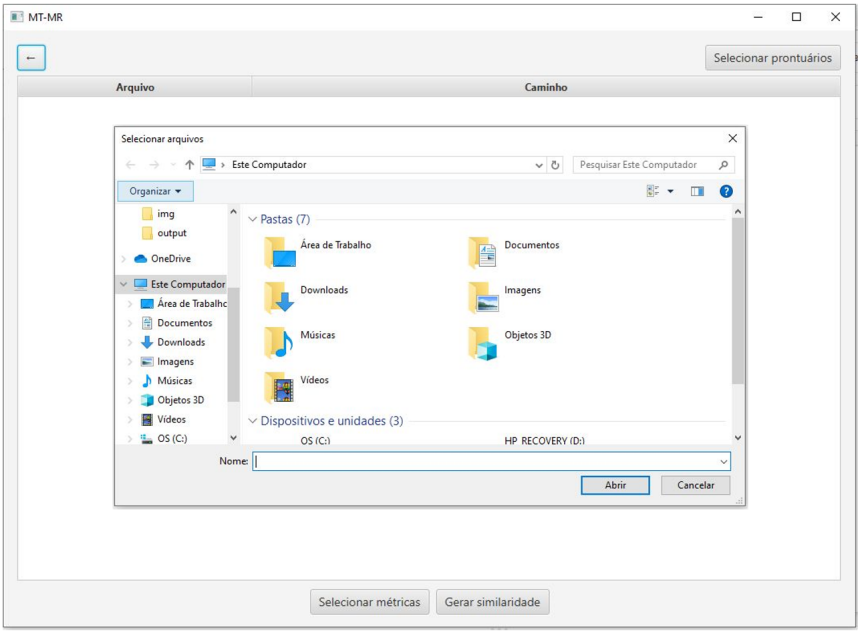
A opção 'Sobre' abre um *Dialog Alert* informando sobre o que é o sistema TM-MR. Semelhantemente, a opção 'Ajuda' abre um *Dialog Alert* informando o passo a passo de funcionamento do sistema. E por fim, a opção 'Iniciar' direciona o usuário à tela de Prontuários, como ilustra a Figura 3.

Figura 3: Tela Prontuários



Na tela de Prontuários, o usuário seleciona o botão 'Selecionar prontuários' e escolhe uma quantidade de prontuários para minerar. Ao escolher os prontuários médicos no formato PDF, o usuário pode escolher 'Selecionar métricas' ou 'Gerar similaridade', como mostra a Figura 4.

Figura 4: FileChooser para escolher os prontuários



Casa o usuário escolha 'Selecionar métricas', o mesmo é direcionado à tela de Métricas onde poderá escolher uma dos quatro algoritmos para obter o resultado da similaridade, como mostra a Figura 6.

Primeiro o usuário seleciona um algoritmo (seta roxa), logo depois o usuário seleciona o botão 'Ver resultados' (seta laranja). Por fim, o usuário possui a opção de 'Exportar resultados' (seta azul), caso queira salvar os dados da *Anamnese* e os resultados de comparação do algoritmo em um novo arquivo TXT, que será salvo na pasta dataset/resultados/.

Figura 5: Opção para selecionar métricas

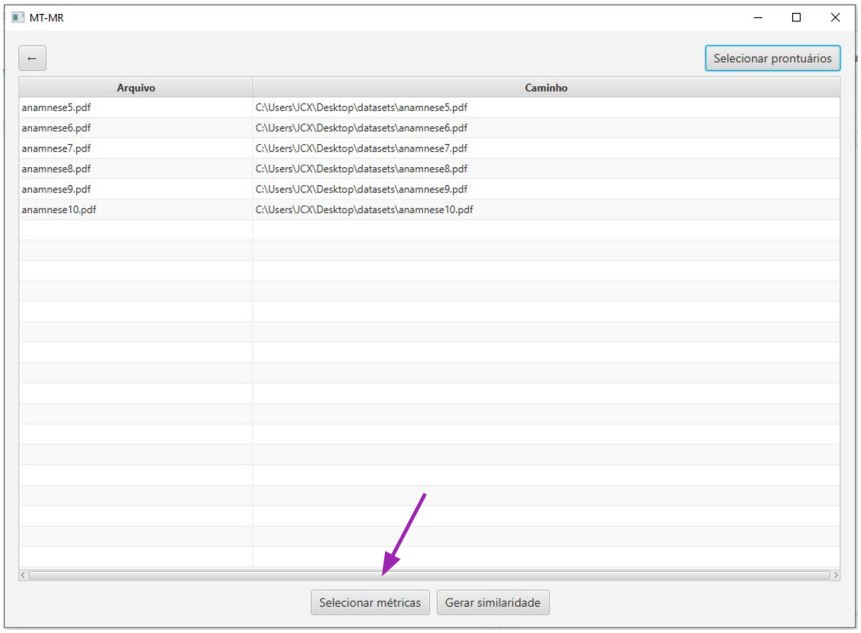
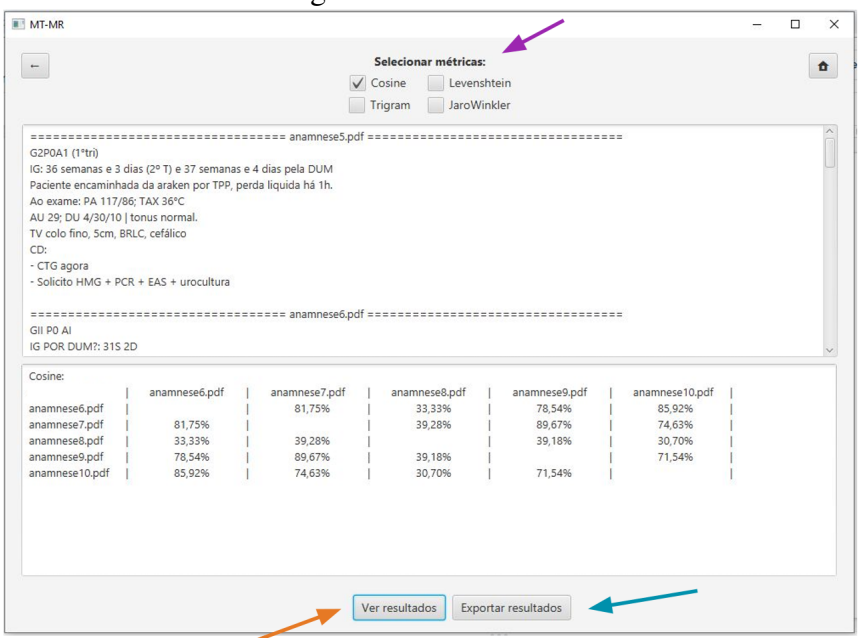
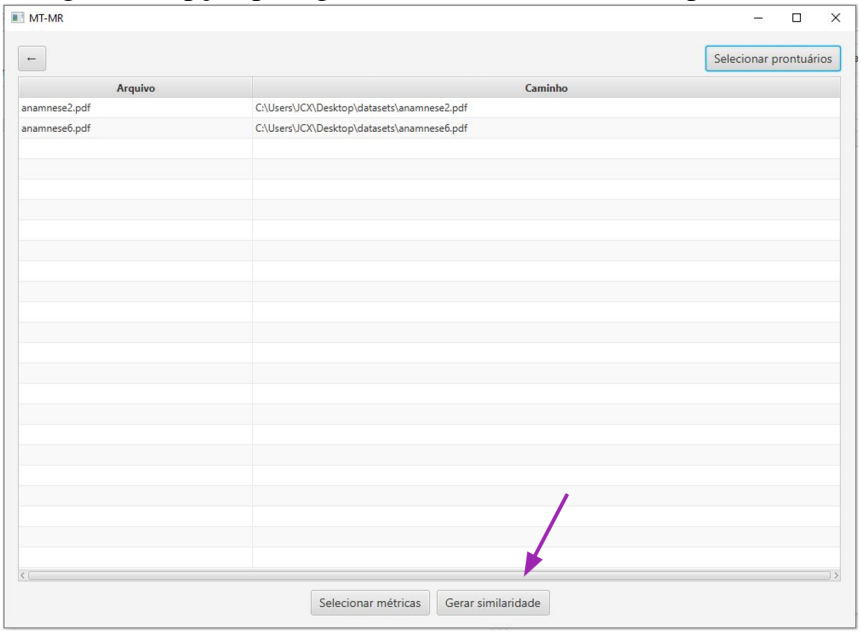


Figura 6: Tela Métricas



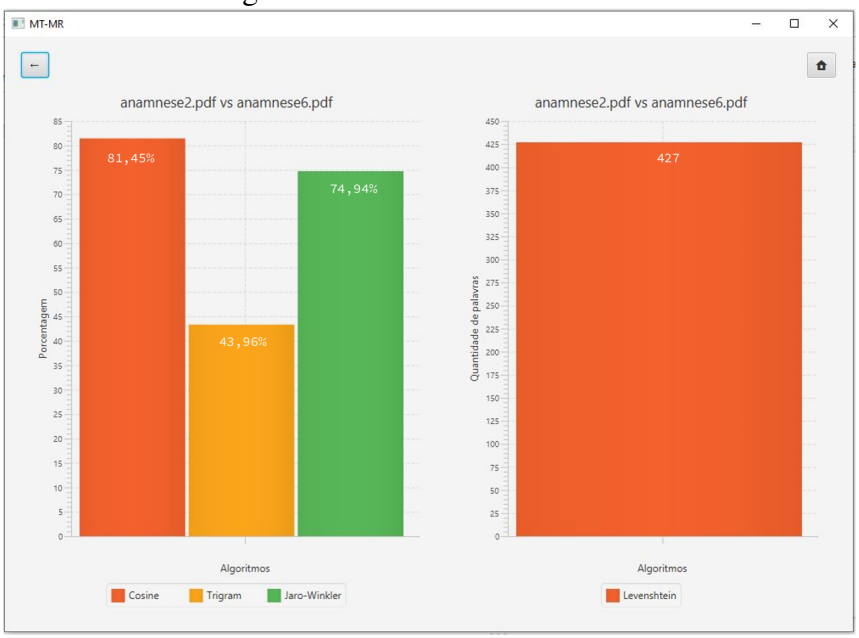
Na Tela Prontuários 3, caso o usuário escolha a opção ’Gerar Similaridade’ (seta roxa na Figura 7), ele deverá escolher apenas 2 (dois) arquivos para tal.

Figura 7: Opção para gerar similaridade entre dois prontuários



Sendo assim, ele será redirecionado à uma nova tela que contém gráficos com os resultados destes dois prontuários, sob os 4 (quatro) algoritmos, como exemplifica a Figura 8.

Figura 8: Gráficos com resultados



4. Experiência

Durante o desenvolvimento da plataforma, houveram experiências positivas e experiências negativas. Alguns dos pontos negativos na experiência, os tópicos que houveram dificuldade na implementação, foram:

- implementação do algoritmo Levenshtein;
- conversão PDF para TXT, devido as dificuldades encontradas nessa etapa, foi utilizado o código pronto do Eugen (disponível em: `https://github.com/eugenp/tutorials/tree/master/pdf`)
- retirar do arquivo txt apenas a seção 'Anamnese', pois foi preciso descobrir o padrão de como o texto ficava depois de convertido;
- utilizar o componente BarChart do JavaFX;

Entretanto, contabiliza-se também os pontos positivos de aprendizagem e conhecimento adquiridos no desenvolvimento, como:

- aprendizagem adquirida no processo e estudo de mineração de dados;
- implementação dos algoritmos Cosine, Trigram e Jaro-Winkler, que acrescentaram conhecimento;
- trabalhar com os componentes do JavaFX, em geral;
- criação e ideias da interface;

5. Conclusão

O sistema TM-MR foi concluído como planejado, atendendo aos seus objetivos específicos e geral, de modo a executar sua função de forma eficiente. Por fim, espera-se que a aplicação cumpra sua finalidade sendo útil no seu funcionamento, visto que seu objetivo foi cumprido. Uma das ideias para trabalhos futuras, seria a construção de um dicionário com as palavras importantes e com peso em significado dos prontuários, para que o sistema possua uma rede de dados significativa para realizar a mineração de texto cada vez mais precisa.