INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Anotação semi-automática de ficheiros de dados

IVO PEREIRA
RUBEN CAFÉ
DUARTE FELÍCIO

Relatório preliminar realizado no âmbito de Projecto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2019-2020

Orientadores : Doutor Nuno Datia
Doutora Matilde Pós-de-Mina Pato

Julho, 2020

Índice

1

Introdução

Nos tempos atuais a quantidade de dados recolhidos e armazenados mundialmente aumenta de dia para dia. É estimado que sejam gerados 4.1 terabytes de dados diariamente por quilómetro quadrado de área urbana [8]. Esta recolha de dados aumentou tanto em volume como em detalhe ao longo do tempo. Os vários fornecedores e/ou agregadores de dados têm diferentes formas de os disponibilizar, originando análises cada vez mais complexas. Em particular, identificar níveis de detalhe e encontrar relações entre os dados tornou-se mais difícil.

É necessário utilizar um processo semi-automático que, sobre um conjunto de dados, gere meta informação para identificar o domínio das variáveis, relações entre os dados e os diferentes níveis de detalhe.

O principal foco deste projeto é desenvolver uma aplicação WEB interativa que, recebendo dados em formato *csv* (*comma separated values*), utiliza um processo de análise semi-automático de modo a proporcionar ao utilizador uma melhor compreensão dos vários níveis de detalhe e relações entre os dados inseridos.

Este processo semi-automático pode, no entanto, revelar falhas no seu funcionamento. Quer sejam devido a erros sintáticos no ficheiro *csv*, ou devido a falha no reconhecimento de níveis de detalhe por parte do algoritmo num dado domínio. Uma das falhas mais comuns é uma das colunas não se encontrar escrita da forma que o algoritmo espera, o que leva a que este não detete essa coluna como sendo um nível de detalhe de outra.

O intuito desta aplicação é expandir o trabalho desenvolvido por Nuno Ribeiro[9], fornecendo ao utilizador uma aplicação web interativa.

Onde após autenticação, é possível carregar ficheiros *csv*, pedir ao algoritmo para os analisar dando hipótese ao utilizador para alterar a informação gerada pelo algoritmo de forma a corrigir eventuais erros, e permitir costumização, para finalmente exportar essa meta informação.

1.1 Problema

Ficheiros *csv* contêm dados organizados em colunas. Dado que este tipo de ficheiros normalmente não contém meta-informações, existe a necessidade de automatizar o processo de obtenção dessa meta-informação, e relações entre dados.

Para análise deste tipo de ficheiros existe a necessidade de identificar os diferentes níveis de detalhe, a conveniência de uma aplicação intuitiva e iterativa, e por fim a utilidade de fornecer ao utilizador a possibilidade de modificar a análise do ficheiro.

Nas aplicações disponíveis para analisar estes ficheiros existe a carência de alguma das funcionalidades apresentadas acima.

1.2 Organização do documento

A organização deste documento divide-se em 4 capítulos. No capítulo 2 são apresentados trabalhos semelhantes ou com objetivos similares ao trabalho que iremos realizar. No capítulo 3 são definidos objetivos e de que forma planeamos cumpri-los. No capítulo 4 é definido e descrito de que forma implementamos o nosso trabalho.

2

Trabalho Relacionado

Neste capítulo é apresentado o estado da arte relativo ao tema que este projeto trata. Na secção 2.1 é apresentada uma dissertação na qual este projeto se baseia. Na secção 2.2 é apresentada uma aplicação com funcionalidades semelhantes que foi analisada para auxiliar na definição de funcionalidades e interface com o utilizador.

2.1 Trabalho relacionado

Nesta secção é apresentada a dissertação do Nuno Ribeiro cuja premissa consiste em analisar semi-automaticamente um ficheiro para identificar o domínio, as relações e os níveis de detalhe das variáveis desse ficheiro.

2.1.1 Dissertação de Nuno Ribeiro

Para obtenção do grau de Mestre em Engenharia Informática, Nuno Ribeiro elaborou a dissertação denominada *Anotação e Extração Semi-Automática de Dados Multidimensionais* [9]. Nela o autor propõe um algoritmo que analisa automaticamente ficheiros csv e obtém meta-informações destes.

Dado o problema de uma crescente quantidade de dados disponibilizados por diferentes fornecedores. Que torna a análise desses dados cada vez mais complexa. O autor propõe uma solução que passa por analisar esses dados fornecidos em formato csv

através de um algoritmo que identifica relações e o domínio de variáveis, assim como níveis de detalhe.

No entanto esta solução ao problema não fornece uma interface com o utilizador iterativa. Nem permite uma costumização por parte do utilizador dos resultados dados pelo algoritmo, como por exemplo alteração de níveis de detalhe de colunas (ou variáveis).

2.2 Aplicações Similares

Nesta secção são apresentados produtos que partilham o objetivo deste trabalho de ser uma ferramenta de análise de dados com uma interface com o utilizador iterativa. Os programas aqui apresentados têm fins monetários e estão no mercado há vários anos e são ferramentas com vastas funcionalidades.

2.2.1 Tableau Prep

O Tableau Prep é uma ferramenta de software para análise e visualização de dados. Esta ferramenta disponibiliza ao utilizador diversas funcionalidades de visualização e análise de dados e tem uma interface com utilizador extremamente intuitiva.

Porém este software não tem a funcionalidade de análise automática para identificação de domínios e níveis de detalhe que pretendemos com o nosso projeto. E, para além disso, é um produto para fins lucrativos, não é *open source*, que o nosso projeto tem o objetivo de ser.

2.2.2 Alteryx

Alteryx é uma aplicação que fornece aos utilizadores a capacidade de preparar e analisar os seus dados, fornecendo a possibilidade de conectar com dados de outras plataformas e juntar esses dados para análise.

Porém com tantas possibilidades e funcionalidades, a interface por vezes pode se tornar confusa e difícil de entender. Este é também um produto que não é *open source* e dispõe de uma grande curva de aprendizagem.

3

Abordagem

3.1 Divisão de funcionalidades e objetivos

O projeto a realizar contém três principais componentes, a componente servidora que trata de receber e tratar pedidos, a componente cliente que trata de apresentar uma interface web interativa ao utilizador e que comunica com a parte servidora e por fim a parte algorítmica de tratamento de ficheiros csv. Logo dividimos estes três focos entre o grupo de modo a ser possível organizar o progresso em use cases em todas as componentes.

3.2 Autenticação

A autenticação de um utilizador lida com dados sensíveis que nunca devem ser comprometidos, no entanto pouco se pode fazer em termos de segurança se o utilizador utiliza uma password simples como *pass123* que pode ser facilmente comprometida através de um simples ataque *brute force* [1]. Logo, o utilizador deve ser forçado a utilizar diferentes tipos de caracteres e um tamanho mínimo de forma a dificultar tais ataques.

O sistema também deve fornecer funcionalidades tais como:

1. Enviar um email de confirmação para confirmar que o email enviado pertence realmente a quem se está a registar;

2. *Two-factor-authentication* de forma a aumentar a segurança e integridade das contas dos utilizadores [6];

3. Alterar a password;

4. Possibilidade de criar uma nova password no eventual esquecimento da password anterior;

5. Alterar o email associado;

6. Possibilidade de fazer download dos dados da sua conta ou de a apagar;

3.3 Carregamento de ficheiros

O carregamento de ficheiros na aplicação pode ser feito por *drag'n'drop* de um ou mais ficheiros numa zona dedicada na aplicação, ou, ao fazer *browse* localmente. O carregamento também pode ser efetuado através de um *URL* que pode ou não provenir de uma página com autenticação. Neste cenário o sistema deve conseguir tratar métodos de autenticação como *OAuth 2.0* [3] pois o sistema não deve pedir a password do utilizar para essa página.

3.4 Ambiente de trabalho

Grande parte das funcionalidades e objetivos desta aplicação situa-se em torno duma *workspace* onde mostra os ficheiros carregados pelo utilizador que têm a sessão iniciada, estes poderão ser organizados pelo tamanho, por nome, pela data em que foram inseridos no sistema ou até mesmo por pastas. Desses ficheiros o utilizador poderá escolher o que será analisado e interagir sobre essa análise, havendo assim a possibilidade de melhorará-la e de importar a meta-data que a própria aplicação gerou.

3.5 Análise de um ficheiro

A análise de um ficheiro tem duas componentes e foram ambas reimplementadas tirando partido da tese de mestrado de Nuno Ribeiro [9]. Estas componentes sendo a sua base de dados - *Knowledge Database* e o algoritmo de análise.

3.5.1 Knowledge Database - Repositório de informação geográfica

A base de dados *Knowledge Database* contém informação geográfica de forma a que a análise consiga perceber os diferentes NUTS presentes nos dados.

3.5.2 Algoritmo de Análise

O algoritmo analisa as diferentes colunas e os próprios valores presentes nas colunas de forma a identificar as diferentes métricas, dimensões e níveis de detalhe presentes. Este algoritmo usa principalmente uma análise aos sufixos e prefixos presentes nos nomes das colunas de forma a identificar os diferentes detalhes.

4

Metodologias

4.1 Tecnologias

As metodologias usadas são focadas em análise de documentos com formato .csv e tratamento de texto, tornando a escolha da linguagem em um aspeto não tão crítico, uma vez que qualquer linguagem têm as capacidades necessárias para implementar as metodologias. No entanto, foi decidido utilizar tecnologias *open-source* de forma a não criar limitações, ou custos adicionais. Foi também dada a preferência a tecnologias que permitam *cross-plataform* para que as nossas aplicações não estejam limitadas quanto às máquinas, e respetivos sistemas operativos, em que podem correr. Para além destas limitações, levámos em consideração a nossa experiência e preferência pessoal. A ferramenta utilizada para desenvolver a aplicação servidor, e que implementa o algoritmo para processamento de dados foi desenvolvida em C#, tecnologia .NET Core.

Para a implementação da base de dados foi usado *SQLServer*. A aplicação web que interage e apresenta os resultados ao utilizador foi desenvolvida usando *JavaScript*, recorrendo à framework *React*.

4.2 *Single Page Application*

Uma *Single Page Application* ou SPA é uma aplicação que funciona dentro de um *browser* e não requer recarregar a página durante o seu uso. Este tipo de aplicações está

presente em várias páginas web tais como *Gmail*, *Google Maps*, *Facebook* ou *GitHub*. Consiste apenas numa página web que carrega todo o seu conteúdo usando *JavaScript* - do qual esta depende bastante. A SPA solicita a marcação e os dados independentemente e faz *render* das páginas diretamente no navegador. Isto é possível graças às estruturas JavaScript avançadas, como *AngularJS, Ember.js, Meteor.js, Knockout.js e ReactJS*. SPA's ajudam a manter o utilizador num espaço confortável, onde o conteúdo é apresentado ao utilizador de maneira simples, fácil e viável [5]. Tendo isto em mente implementámos a nossa aplicação como SPA.

4.3 Autenticação - .NET Core Identity

De forma a implementar a autenticação de maneira segura utilizámos o modelo *Identity* oferecido pela tecnologia .NET Core [2]. Este modelo suporta funcionalidades de *login* tais como utilizadores, passwords, dados de perfis, papéis, *tokens* de autenticação, confirmação de email, *Two-factor-authentication* [6] e mais.

4.4 Carregamento de ficheiros

A nossa aplicação permite upload de ficheiros diretamente da máquina do utilizador, ou através de um endereço fornecido pelo utilizador de modo a que a aplicação vá efectuar o download do ficheiro a esse endereço.

Qualquer ficheiro carregado na aplicação é guardado no *file system* do servidor após serem efectuadas algumas verificações. Os ficheiros são guardados com um nome aleatório gerado pela aplicação de forma a proteger contra ataques maliciosos. Todos os ficheiros de um utilizador são guardados numa pasta cujo nome é o id desse utilizador. Na nossa base de dados mapeamos os nomes aleatórios com os nomes originais, assim como guardamos algumas meta-informações. sobre cada ficheiro, tais como o tamanho, a origem do ficheiro, etc.

4.4.1 Frontend - *React Dropzone*

A parte cliente para carregamento de ficheiros foi implementada através da biblioteca *React Dropzone* que facilita a criação e customização de uma zona de *drag'n'drop* para ficheiros csv [4].

4.4.2 Backend

Numa das opções para upload de ficheiro o servido recebe um ficheiro proveniente do cliente em formato *multipart*. É aberto um *stream* com o conteúdo que vem no pedido, *stream* esse que é guardado no servidor como ficheiro. É utilizado um *stream* ao invés de um *buffer* porque a nossa aplicação aceita ficheiros de grande dimensão e um *buffer* pode utilizar demasiada memória [7].

Como segunda opção o servidor recebe no pedido o *uri* do ficheiro que tem de efectuar o download. São feitas a verificações se é um ficheiro aceitável pela aplicação e de seguida é aberto um *stream* desse endereço e guardado como ficheiro no servidor.

Referências

[1] What's a brute force attack? https://www.kaspersky.com/resource-center/definitions/brute-force-attack. Accessed: 01-05-2020.

[2] Introduction to identity on asp.net core. https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-3.1&tabs=visual-studio. Accessed: 01-05-2020.

[3] Oauth 2.0. https://oauth.net/2/. Accessed: 01-05-2020.

[4] react-dropzone. https://github.com/react-dropzone/react-dropzone. Accessed: 01-05-2020.

[5] Single-page application vs. multiple-page application. https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58. Accessed: 01-05-2020.

[6] What are the benefits of two-factor authentication? https://messente.com/blog/most-recent/benefits-of-two-factor-authentication. Accessed: 01-05-2020.

[7] Upload files in asp.net core. https://docs.microsoft.com/en-us/aspnet/core/mvc/models/file-uploads?view=aspnetcore-3.1. Accessed: 08-04-2020.

[8] Ciprian Dobre and Fatos Xhafa. Intelligent services for big data science. Future generation computer systems, 37:267–281, 2014.

[9] Nuno Ribeiro. Anotação e Extração Semi-Automática de dados multidimensionais. Master's thesis, Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia, Portugal, 2019.