UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

IMPACTO DE OBRAS LITERÁRIAS NAS REDES SOCIAIS

Carlos André Freitas Barata

PROJETO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

2014

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

IMPACTO DE OBRAS LITERÁRIAS NAS REDES SOCIAIS

Carlos André Freitas Barata

PROJETO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

Dissertação orientada pelo Prof. Doutor Francisco José Moreira Couto
e co-orientado pelo Prof. Doutor Tiago João Vieira Guerreiro

2014

Agradecimentos

Em primeiro, lugar agradeço ao Sapo Labs pelo contrato de estágio oferecido, o que permitiu esta tese ser possível. Agradeço em particular a Pedro Torres, Jorge Teixeira, Benjamin Junior, Ana Gomes e Bruno Tavares pela grande disponibilidade e ajuda ao longo de todos os projetos em que colaborámos.

Agradeço aos meus orientadores Francisco Couto e Tiago Guerreiro pelo apoio e orientação dados ao longo da tese e em outros projetos para além confiança depositada em mim.

Não posso deixar de agradecer aos meus pais por não me darem o que eu quero mas o que eu preciso, por sempre me mostrarem qual o melhor caminho a seguir e lutar para me oferecer condições para ultrapassar os obstáculos, apesar de todas as dificuldades.

A todas as pessoas que tiveram envolvidas nos projetos "O Mundo em Pessoa", "Lusica", "Onde há bola", "Missinks" e "SocialBus" pelo bom trabalho em equipa que permitiu ter sempre os projetos prontos dentro dos prazos definidos.

Agradecimentos especiais a todos os meus amigos que tiveram presentes em vários momentos da minha vida e que estiveram do meu lado nos bons e maus momentos. Em especial agradeço aos meus velhos amigos João Martins, Diogo Santos e Tânia Sanches por me aturarem há tanto tempo e estarem sempre dispostos para ajudar em qualquer situação. Agradeço ainda a todas as pessoas que me ajudaram no meu percurso na FCUL, em especial Rita Henriques, Tiago Aparício, Fábio Santos, José Carilho, João Nascimento, Gonçalo Semedo, Mónica Abreu, Rafael Oliveira e Luís Rochinha pelo bom trabalho em equipa em várias ocasiões, pela disponibilidade para ajudar quando eu mais precisava, pela diversão proporcionada e principalmente pela boa disposição, que está sempre garantida.

Para meus pais e amigos

Resumo

As redes sociais são, hoje em dia, o maior meio de partilha de informação. Tornaram-se ao longo dos anos ferramentas fundamentais no dia-a-dia dos utilizadores da internet. Da vasta informação partilhada nestas redes sociais existem citações a obras artísticas que na maior parte das vezes não são referenciadas. Geralmente, os utilizadores das redes sociais que partilham uma citação, colocam o nome do autor mas não existe referência à que obra pertence.

Outro problema com a partilha deste tipo de informação é que as citações nem sempre estão iguais ao original, sendo que os utilizadores utilizam sinónimos ou calão, o que torna um desafio a comparação da informação. Para além disto, existem citações referentes a um autor que na realidade pertencem a outro, o que dá origem a informação falsa.

Nesta tese foi criada uma abordagem para fazer corresponder mensagens partilhadas nas redes sociais com a obra de um determinado autor. Para concretizar esta abordagem foi, inicialmente, construído um projeto chamado "O Mundo em Pessoa" que, através da recolha de citações das redes sociais Twitter e Facebook, faz o mapeamento destas citações com a obra original do poeta Fernando Pessoa. A partir deste projeto foi criada uma arquitetura, o "Social Impact". O "Social Impact" tem como principal objetivo a abstração da estrutura de "O Mundo em Pessoa", contando com a ajuda de um sistema de *information retrieval* concretizado através da ferramenta Apache Lucene e com o uso do projeto SocialBus para a recolha de mensagens partilhadas nas redes sociais.

A partir do "Social Impact" foi criada uma nova versão melhorada de "O Mundo em Pessoa" e o "Lusica", um projeto que tem como objetivo mostrar o que é falado nas redes sociais sobre os artistas musicais lusófonos. Para além destes dois casos de estudo do "Social Impact", foram também reutilizados componentes desta arquitetura para outros projetos: "Onde há bola", projeto que mostrava os locais onde assistir aos jogos do campeonato do mundo de futebol de 2014, e o "Missinks" que compara os resultados das pesquisas do Google de dois domínios diferentes.

Palavras-chave: redes sociais, pesquisa em texto, prospecção de dados, correspondência de texto, recuperação de informação

Abstract

Nowadays, social networks are the biggest information sharing environment. They have become fundamental tools, to the web users, over the years. From the wide amount of information shared in these social networks, there are quotes to art works that most of the times are not referenced. Usually, the social network users who share quotes put the name of the author but there is no reference to the art work that this quote belongs.

Another problem with the sharing of this kind of information is that the quotes are not always equal to the original quotes, wherein the users use synonyms or slang words making this a challenge to compare information. Besides that, there are quotes related to one author but in fact, the quote belongs to another author which origins false information.

In this thesis was created one approach to match shared messages in social networks to the original art work from a particular author. To achieve this approach, we built one prototype called "O Mundo em Pessoa" that through the gathering of quotes from Twitter and Facebook, maps quotes to original art works from the poet Fernando Pessoa. We created one architecture, called "Social Impact", that provides an abstraction of the prototype "O Mundo em Pessoa". Based on the "Social Impact" and on the information retrieval tools, like Apache Lucene and SocialBus, was created a new and more efficient version of "O Mundo em Pessoa". The same strategy was also adopted for "Lusica", a prototype that gives an overview of what is being said about Portuguese songs in the social networks. In addition, to these two case studies, we use components of the "Social Impact" arquitecture to create another two projects: "Onde há bola", project that shows the places where to watch the football games of the world cup 2014; and "Missinks", a tool that compares the results of the searches in the Google domains of two different countries.

Keywords: social networks, text search, web data mining, text matching, information retrieval

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

As redes sociais que foram surgindo ao longo da última década mudaram a maneira como se comunica, tornando-se ferramentas fundamentais nas relações humanas, visto que está à distância de um clique a possibilidade de partilhar conteúdos e enviar mensagens. Por consequente, tornou-se não só o meio de comunicação e partilha de informação mais popular, como também um objeto de investigação bastante atrativo em várias áreas tais como extração e análise de informação, como Kwak et al. [2010] refere.

1.1 Motivação

A informação que é partilhada nas redes sociais pode ter vários tipos e formatos, sendo a sua grande maioria mensagens escritas das quais, muitas são citações a frases ou obras de algum autor. Conseguimos facilmente identificar uma citação quando é feita uma referência ao nome do autor, contudo não existe correspondência a que texto, livro ou obra desse autor a citação pertence.

As redes sociais criaram mecanismos de forma a permitir aos utilizadores colocarem referências nas suas mensagens como é o caso das *hashtags*. Estes mecanismos têm uma baixa taxa de utilização como mostra o estudo feito por Weerkamp et al. [2011]. Este estudo concluí que na rede social Twitter, a língua que mais usa *hashtags* é a alemã e apenas 25% dos seus *tweets* os utilizam.

Para além do problema de identificar a referência às obras numa mensagem, muitas das citações partilhadas nas redes sociais não estão corretas ou não estão exatamente como a obra original pelo uso de sinónimos ou mesmo por erros gramaticais. Muitas vezes as citações são associadas a um artista que na verdade pertence a outro (por exemplo: "Pedras no caminho? Guardo todas, um dia vou construir um castelo…", é comum esta frase estar associada a Fernando Pessoa quando na verdade é de Augusto Cury).

1.2 Objetivos

O principal objetivo desta tese é criar uma abordagem que ajude a mapear mensagens partilhadas nas redes sociais com uma determinada obra artística, analisando o conteúdo das mensagens e comparando-o com a obra do autor citado. Para ajudar a alcançar este objetivo geral foram definidos os seguintes sub-objetivos:

1. Recolha automática das obras de um determinado autor.
2. Recolha automática das mensagens partilhadas nas redes sociais que citem as obras.
3. Construção de um sistema que faça o mapeamento das obras com as citações recolhidas, referidas nos pontos anteriores.
4. Implementação de dois casos de estudo de forma a dar um propósito à abordagem. Os dois casos de estudo servirão não só para tornar a abordagem aberta a vários contextos como também contribuir para uma análise de performance e deteção/correção de possíveis erros. Para além disto, estes casos de estudo ajudarão na divulgação das obras de autores lusófonos.
   (a) "O Mundo em Pessoa", projeto de recolha automática de citações de Fernando Pessoa (ortónimo e heterónimos) a partir das redes sociais. Este projeto tem como objetivo tentar perceber quais os versos e frases de Fernando Pessoa que mais inspiram os leitores de todo o mundo e também, conduzir todos aqueles que usam as palavras de Pessoa até ao seu texto original, ampliando o número de leitores e o conhecimento da sua obra.
   (b) "Lusica" que à semelhança de "O Mundo em Pessoa", é um projeto de recolha automática de citações de música de artistas lusófonos a partir das redes sociais, cujo propósito será a aplicação de um novo contexto à abordagem, neste caso o contexto será a área musical.

1.3 Contribuições

As principais contribuições desta tese foram:

1. Implementação do primeiro protótipo de "O Mundo em Pessoa"[1].
2. Implementação da arquitetura "Social Impact" que faz deteção de citações nas mensagens das redes sociais e pode ser estendida para vários contextos diferentes.
3. Implementação do segundo protótipo de "O Mundo em Pessoa" utilizando a arquitetura "Social Impact", ou seja, um caso de estudo para o "Social Impact".

[1] Link para "O Mundo em Pessoa": `http://fernandopessoa.labs.sapo.pt/`

4. Implementação do primeiro protótipo do "Lusica"[2] utilizando a arquitetura "Social Impact", ou seja, outro caso de estudo para o "Social Impact".

5. Colaboração entre SAPO Labs e Faculdade de Ciências da Universidade de Lisboa na construção de aplicações *web* com um grande impacto na visibilidade nos meios de comunicação social para ambas as instituições.

6. Exposição e apresentação dos casos de estudo nos eventos Sapo Codebits VII[3] e no Dia Aberto 2014[4].

7. Implementação do "Missinks"[5] que reutiliza elementos da arquitetura "Social Impact".

8. Orientação no projeto "Onde há Bola"[6] que reutiliza elementos da arquitetura "Social Impact".

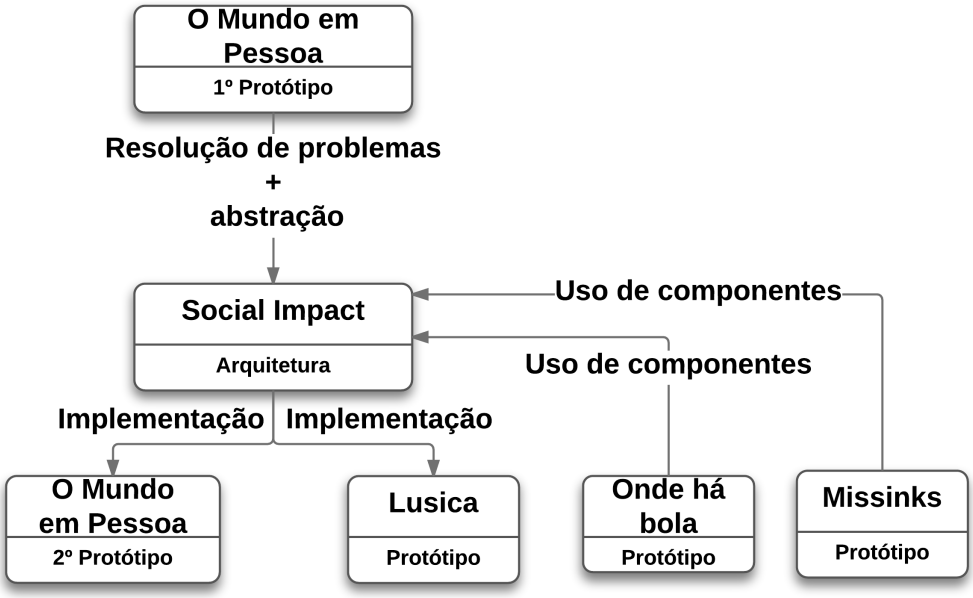

Figura 1.1: Visão geral da tese

Em suma, a Figura 1.1 mostra todas as contribuições desta tese e como se relacionam entre elas. Em primeiro lugar foi implementado o primeiro protótipo em colaboração com o Sapo Labs que foi lançado na data do aniversário dos 125 anos de Fernando Pessoa, e por isso foi um protótipo muito rudimentar com o principal objetivo de cumprir este prazo.

[2]Link para o "Lusica": `http://www.lasige.di.fc.ul.pt/webtools/lusica/`
[3]Link para o Codebits: `https://codebits.eu/`
[4]Link para o Dia Aberto: `http://www.fc.ul.pt/pt/pagina/1932/dia-aberto-2014`
[5]Link para o "Missinks": `http://missinks.fc.ul.pt/`
[6]Link para o "Onde há Bola": `http://ondehabola.labs.sapo.pt/`

Foram portanto detetados vários problemas e haveria a necessidade da criação de uma arquitetura abstrata que fosse extensível a outros contextos, foi por isso criado o "Social Impact". A partir do "Social Impact" foi implementada uma nova versão de "O Mundo em Pessoa" e a primeira versão do "Lusica". Para além destes projetos, foram criados mais dois que não são uma implementação direta mas que reutilizaram componentes do "Social Impact", que foi o "Onde há Bola" e o "Missinks".

1.4 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 1 (Introdução) é um capítulo introdutório que está dividido em quatro secções. A primeira apresenta as principais motivações para o desenvolvimento desta tese, a segunda os objetivos que se pretende cumprir, na terceira as principais contribuições e por último, a estrutura este documento.

- Capítulo 2 (Trabalho relacionado) apresenta o contexto onde esta tese se insere e alguns trabalhos realizados nesta área. Este capítulo está dividido em três secções: *Web data mining* onde são apresentados os métodos de recolha e análise de informação na web, *Information Retrieval* que apresenta as técnicas existentes para a obtenção de informação de uma vasta coleção de dados, e por fim, Projetos e ferramentas relacionadas que apresenta os projetos e ferramentas utilizados para a implementação desta tese e que usam as técnicas descritas nas Secções anteriores.

- Capítulo 3 ("Social Impact") apresenta a solução proposta para cumprir o objetivo descrito na Secção 1.2. Está subdividida em duas secções: "O Mundo em Pessoa" que apresenta o primeiro protótipo construído apenas focado no contexto das obras literárias, Requisitos que apresenta os requisitos funcionais e não funcionais do sistema "Social Impact" e Arquitetura que apresenta a especificação da arquitetura "Social Impact" e a sua implementação.

- Capítulo 4 (Resultados) apresenta os sistemas contruídos a partir da arquitetura "Social Impact" descrita no Capitulo 3. Está divido em 4 secções: Caso de estudo: "O Mundo em Pessoa", que apresenta o projeto como caso de estudo do "Social Impact", Caso de estudo: "Lusica" que apresenta outro caso de estudo mas com o contexto musical, "Onde há bola" que apresenta um projeto que reutiliza componentes da arquitetura e "Missinks" que apresenta outro projeto que reutiliza componentes da arquitetura.

- Capítulo 5 (Conclusão) apresenta um sumário do trabalho desenvolvido, resultados obtidos e contribuições.

Capítulo 2

Trabalho relacionado

2.1 *Web data mining*

O rápido crescimento da *world wide web* na última década faz com que esta seja a maior fonte de dados do mundo. Tornou-se a plataforma mais importante e popular não só para encontrar informação mas também para fornecer serviços. Todos os autores de livros e artigos que abordam *web mining* Liu [2007], Wang [2000], Akerkar et al. [2012], concordam com o facto de que este rápido crescimento faz com que, por vezes, a informação não se encontre estruturada. Esta é a grande motivação para a área de *web data mining* visto que, de forma a encontrar a informação pretendida nesta grande fonte de dados, será fundamental a implementação de mecanismos de *data mining* na *web*.

Liu [2007] apresenta uma lista de características únicas que fazem com que seja fundamental a prospeção de informação na *web* de forma a obter os dados e conhecimento pretendidos:

- A quantidade de informação é enorme e sempre em crescimento. A cobertura da informação é grande e diversa.
- Tipos de dados na internet podem ser variados e podem, ou não, estar estruturados.
- A informação é heterogénea, ou seja, várias páginas contêm a mesma informação, podendo conter diferentes palavras ou formatos, tornando assim a integração de informação um problema complexo.
- Muita informação na internet está relacionada através de *hyperlinks* que pode completar informação de uma determinada página.
- Muita da informação contida numa página *web* não é relevante para uma determinada aplicação (links de navegação, publicidade, etc). Para a prospeção da informação relevante é necessário remover este tipo de informação não relevante. Para além disto a informação pode ser de baixa qualidade, informação errada ou enganadora visto que qualquer utilizador pode inserir informação numa página *web*.

- A *web* é utilizada não só para consultar informação como para aceder a serviços como, por exemplo, comprar produtos, pagar contas, etc.

- A informação é dinâmica, ou seja, pode estar sempre a mudar. A monitorização das alterações de informação pode ser importante para várias aplicações

- A *web* é uma sociedade virtual. Não é apenas dados mas interações entre pessoas, organizações e sistemas automatizados.

Estas características apresentam não só desafios mas também oportunidades para a prospeção de dados e descobrir informação e conhecimento através da internet. De forma a explorar *web mining* é necessário perceber *data mining*. A subseção seguinte irá descrever estas duas técnicas e mostrar as suas diferenças e semelhanças.

2.1.1 *Data mining versus web mining*

Data mining é um processo de descoberta de padrões e conhecimento através de fontes de dados (bases de dados, textos, etc). É uma disciplina que envolve várias áreas como *Machine learning*, Estatística, Inteligência artificial e visualização.

Segundo Han et al. [2006], *data mining* atraiu grande atenção na indústria da informação e na sociedade como um todo, nos últimos anos, durante a disponibilidade de grande quantidade de dados e a necessidade iminente de tornar estes dados em informação útil e em conhecimento. A informação recolhida através de um sistema de *data mining* poderá ser útil para, por exemplo, análises do mercado, deteção de fraudes, controlo de produção e exploração científica.

Liu [2007] descreve o processo de um sistema *data mining* começando com uma compreensão do domínio da aplicação, identificando as fontes de dados e os dados necessários. Depois desta compreensão é procedido à recolha dos dados. Aos dados recolhidos deste processo serão aplicados 3 passos:

- Pré-processamento – É necessário remover os dados não relevantes. Os dados podem ser demasiado grandes e ter atributos irrelevantes, pelo que se torna necessário este processo de redução de dados.

- *Data-mining* – os dados pré-processados são passados por um algoritmo de *data mining* que produz padrões ou conhecimento através dos dados.

- Pós-processamento – De todos os padrões gerados é necessário identificar aqueles que são úteis através de técnicas de avaliação e visualização. Este processo é quase sempre iterativo. O processo é iterado até produzir resultados satisfatórios que são incorporados no sistema.

Web mining tem o mesmo objetivo e foi criado pela mesma motivação que o *data mining*. A principal diferença prende-se com o facto de que a descoberta da informação ou conhecimento provém da *web* através de *hyperlinks* estruturados, conteúdos de páginas *web* ou dados de usabilidade dessas mesmas páginas. Segundo a maior parte dos autores Liu [2007], Wang [2000], Akerkar et al. [2012], as categorias de *web mining* são baseadas no tipo de dados que existe na *web*:

- *Web structure mining* - descobre conhecimento útil através de *hyperlinks* que representam a estrutura da *web*. Por exemplo, através dos *links* conseguimos descobrir páginas *web* importantes, o que é bastante usado para motores de busca. Também podemos descobrir, por exemplo, comunidades de utilizadores que partilham interesses comuns. O *data mining* tradicional não contém esta tarefa porque não costuma existir uma estrutura de *hyperlinks* na sua informação.

- *Web content mining* - extrai informação ou conhecimento útil a partir do conteúdo das páginas *web*. Por exemplo, conseguimos automaticamente classificar páginas *web* de acordo com os seus tópicos. Esta tarefa é parecida com a de *data mining* tradicional, contudo conseguimos descobrir padrões nas páginas *web* e extrair informação útil como descrições de produtos, publicações em fóruns, etc.

- *Web-usage mining* - para descobrir padrões de acesso dos utilizadores a partir de *logs* de acesso que gravam cada *click* feito por cada utilizador.

No *data mining* tradicional, o processamento dos dados estão muitas vezes já contidos e guardados num *data warehouse*, ao passo que no *web mining*, os dados podem ser uma tarefa substancial, especialmente para *web content mining* o que pode envolver o uso de *crawlers* a um grande número de páginas *web*. Aos dados recolhidos, será aplicado o mesmo processo de *data mining*: pré-processamento, *web mining* e pós-processamento. Contudo, as técnicas usadas para cada passo são diferentes das usadas no tradicional. Assim, *web mining* não é uma aplicação de *data mining* visto que a estrutura e heterogeneidade da informação do *web mining* pode ser bastante diferente de *data mining* contudo, *web mining* usa as técnicas de *data mining* de forma a processar a informação. Neste documento apenas irão ser descritos os dois primeiros tipos de *web mining* visto que, de acordo com os objetivos descritos na Secção 1.2, não será tido em conta o acesso do utilizador a um determinado website como é previsto no *web usage mining*.

2.1.2 *Web structure mining*

Liu [2007], os motores de busca antigos devolviam páginas relevantes para os utilizadores baseando-se apenas na similaridade do conteúdo. Contudo tornou-se claro que a similaridade do conteúdo apenas não era suficiente por duas razões:

1. O número de páginas *web* cresceu rapidamente. Dada uma consulta, o número de páginas relevantes poderia ser enorme. Por exemplo, dado um conjunto de termos de procura como "*classification technique*", o motor de busca Google estima que existam por volta de 10 milhões de páginas relevantes. Esta abundância de informação causa um grande problema de *ranking*, como por exemplo, como escolher apenas 30 ou 40 páginas que se vão apresentar em primeiro lugar ao utilizador.

2. A similaridade do conteúdo pode facilmente ser alvo de *spam*. Um dono de uma página pode repetir palavras importantes e adicionar remotamente palavras relacionadas para que as suas páginas apareçam mais à frente nos *rankings* ou para fazer com que as páginas se tornem relevantes num grande número de possíveis consultas. Por exemplo, a palavra "cruise" tem um ranking elevado por causa da grande procura pelo nome do ator Tom Cruise e pode ser utilizada, por exemplo, por agências de viagens de forma a aparecerem mais à frente nos rankings.

A solução encontrada para estes problemas foram os *hyperlinks*. Ao contrário dos documentos de texto que são considerados independentes, as páginas *web* são ligadas através de *hyperlinks*, que carregam informação importante.

Os *hyperlinks* podem ser usados para organizar uma grande quantidade de informação dentro de um página *web* ou apontar para outras páginas *web*, o que indica um transporte de autoridade para as páginas que estão a ser apontadas. Neste último caso, as páginas apontadas poderão conter informação que completa a informação apresentada pela página que contém esta referência.

Web structure mining tem como objetivo identificar a relação entre páginas *web* ligadas por similaridade de informação ou por *hyperlinks*. A partir de um *hyperlink* é gerada a informação da estrutura da página. Isto permite, por exemplo, a um motor de busca extrair dados relativos a uma pesquisa diretamente do conteúdo de uma página. Os programas que fazem esta pesquisa e extração dos dados através de *hyperlinks* são conhecidos como *web crawlers*.

Chakrabarti [2003] considera o principio básico dos *crawlers*, a recolha de informação de uma página recolhendo também *hyperlinks* para outras páginas que não foram recolhidas, ou seja, quando uma página é recolhida, são identificados os *hyperlinks* para outras páginas que representam potencial trabalho pendente para o *crawler*. Apesar de um *crawler* estar sempre à procura de novas páginas, não há garantia que todas as páginas acessíveis serão encontradas por esse *crawler*.

Os exemplos de *web crawlers* mais conhecidos são os motores de busca que percorrem a *web* processando e indexando as páginas recolhidas para serem apresentadas quando um utilizador faz uma pesquisa. O motor de busca mais popular é o Google, cujos autores Brin et al. [1998] apresentaram a primeira geração de *crawlers* deste motor de busca. Os resultados da avaliação desta ferramenta foram 26 milhões de pedidos HTTP

em 9 dias. Para além dos números foi também verificada a qualidade dos resultados de acordo com a relevância que tinham para a pesquisa. Concluiu-se portanto através dos exemplos apresentados que esta era uma ferramenta com resultados bastante relevantes para as procuras.

Posteriormente foi lançado o Mercator por Heydon et al. [1999] que é um *crawler* com o objetivo de indexar a intranet de organizações. Este *crawler* obteve uma média de 122 documentos por segundo em 8 dias e efetuando cerca de 80 milhões de pedidos HTTP. Comparando com o do Google, teve resultados bastante favoráveis tendo como principal fator a indexação não só de páginas HTML, como é o caso do Google, mas também do tipo MIME.

Mais recentemente foi lançado UbiCrawler por Boldi et al. [2004] que é um *web crawler* escalável, tolerante a faltas e totalmente distribuído. O resultado foi um sistema independente de plataforma, linear, escalável, resistente a faltas e a com descentralização de cada tarefa.

Conclusão, *Web structure mining* tem uma relação natural com o *web content mining*, visto que é bastante provável que um documento *web* contenha *hyperlinks* no seu conteúdo que serão importantes para encontrar novas páginas. É portanto bastante frequente que uma aplicação de *web data mining* combine estas duas categorias, como refere Wang [2000].

2.1.3 *Web content mining*

Web content mining é a recolha de informação automática de texto, imagens, áudio, vídeo, metadados e *hyperlinks* de uma página *web*. A procura destas páginas *web* é feita através do *structure mining* que fornece resultados baseados no nível de semelhança com a consulta feita.

Segundo Wang [2000], os dados que se encontram nas páginas web podem ser semi-estruturados (por exemplo: documentos HTML), estruturados (se os dados provierem de uma tabela) ou não estruturados (o que acontece na maioria dos dados).

São identificados dois desafios no *web content mining*: a extração da informação e depois da extração feita, a integração com a informação que já foi recolhida.

Extração de informação

Um programa que extrai dados da web é chamado *wrapper*. Os dados estruturados na *web* são tipicamente dados guardados em bases de dados que posteriormente são devolvidos e disponibilizados pelas páginas *web*. Extrair estes dados é bastante útil visto que é possível obter e integrar várias fontes para fornecer serviços com informação extra como, por exemplo, informação *web* personalizável de várias fontes, comparação de preços de um produto em várias fontes, etc.

Com cada vez mais organizações e instituições a divulgarem informação na *web*, a capacidade de extrair estes dados está a tornar-se cada vez mais importante. Existem 3 abordagens de extração de informação na *web*:

- Abordagem manual: O programador observa a página *web* e o seu conteúdo de forma a encontrar alguns padrões e de seguida escreve um programa para extrair os dados. Para tornar o processo mais simples para os programadores várias linguagens com especificações padrão e interfaces com o utilizador foram criadas. Apesar da facilidade de implementação, esta aproximação não é escalável para um grande número de páginas *web*.

- *Wrapper induction* – Esta é uma aproximação semi-automática onde é usado um conjunto de regras de extração aprendidas através de uma coleção de páginas classificadas manualmente. As regras são depois usadas para extrair dados de páginas formatadas similarmente.

- Extração automática – Dadas as páginas, automaticamente encontra padrões ou gramáticas das páginas de forma a extrair os dados. Esta abordagem elimina o esforço manual de classificação e pode ser escalada para um número de páginas maior.

O trabalho efetuado e publicado na área é vasto. Por exemplo, a ferramenta ANDES por Myllymaki [2002], ajuda a resolver os problemas identificados na extração de dados na web: *forms* HTML e Javascript que geram informação automaticamente. Estes elementos tornam um desafio a recolha desta informação, *datasets* incompatíveis e dados em falta ou dados conflituosos. De forma a ultrapassar estes problemas foi necessária a construção de um sistema de validação de dados e recuperação de erros de forma a lidar com falhas na extração dos dados.

O artigo de Kushmerick [2000] fala da técnica *wrapper induction* de forma a construir um *wrapper* semiautomático. Com este estudo chegam-se a duas conclusões: é necessário um sistema de aprendizagem automático para manter a grande quantidade de bibliotecas dos *wrappers* visto que estão constantemente a aparecer novas fontes na *web* e a avaliação feita na ferramenta revelou algumas falhas em relação a expressividade e a eficiência.

Integração de informação

Os dados extraídos das páginas *web* são geralmente guardados em bases de dados. Contudo, para uma aplicação pode não ser suficiente extrair dados de uma única fonte, por exemplo, para oferecer um serviço com informação mais enriquecida. Neste caso, para além da extração é necessária a integração da data extraída de forma a produzir uma base de dados consistente e coerente.

Integração significa combinar diferentes colunas em diferentes tabelas que contêm o mesmo tipo de informação e combinar valores que são semanticamente idênticos mas que são representados de forma diferente em diferentes páginas *web*. A integração da informação é uma área de pesquisa bastante limitada sendo mais focada em *Web query interfaces* visto que muitos dos problemas destas duas áreas são bastante semelhantes.

2.1.4 Redes Sociais

A web evoluiu radicalmente de um estado estático, de apenas documentos ligados por *hyperlinks*, para um estado interativo de forma a potenciar a colaboração dos seus utilizadores. Desta forma a web tornou-se numa plataforma de geração, circulação e difusão de conteúdos.

Este estado é conhecido como *web 2.0*, nome que foi dado pela empresa americana O'Reilly Media em 2004, através de Romaní et al. [2009], para designar esta segunda geração de comunidades e serviços. Nesta nova geração, os utilizadores tornaram-se mais ativos na geração e regulação de conteúdos como por exemplo: *blogs*, *wikis*, redes sociais, etc. De todos os serviços criados com o aparecimento da *web 2.0*, os que mais se destacam são as redes sociais.

As redes sociais têm aumentado drasticamente a comunicação e interação na *web* pois permitem que que milhões de utilizadores partilhem as suas opiniões numa grande variedade de tópicos. Consequentemente o uso e integração de redes sociais está a tornar-se cada vez mais importante para empresas e instituições. A maioria das redes sociais já fornecem APIs que permitem aceder aos conteúdos partilhados pelas suas ferramentas de forma a melhorar a personalização de uma aplicação que explora estes dados.

O trabalho nesta área é enorme visto a importância que esta área está a ter na web. Leite et al. [2013] descreve uma aplicação destinada a monitorizar o impacto de notícias de um determinado jornal nas redes sociais Twitter e Facebook. Esta aplicação foi construída com base nas APIs que as redes sociais disponibilizam e no final conseguiram cumprir o objetivo com uma ferramenta de fácil uso por parte dos utilizadores.

Outro trabalho é o descrito na tese de mestrado de Oliveira [2010] e no artigo de Boanjak et al. [2012] onde é descrito o sistema TwitterEcho. O TwitterEcho é um sistema que recolhe "tweets" da rede social Twitter de um grupo restrito, a comunidade portuguesa. O "TwitterEcho" disponibiliza através de um termo todas as citações a esse termo que estão a ser referidas nesta comunidade.

2.2 *Information Retrieval*

Information retrieval é a tarefa de obter informação relevante para uma pesquisa numa vasta coleção de documentos. A unidade básica de *information retrieval* é o documento e portanto a sua base de dados é uma grande coleção de documentos.

Na prática, *information retrieval* baseia-se em encontrar um conjunto de documentos que são relevantes para a procura efetuada. Desta procura é gerado um ranking com os documentos ordenados pela sua relevância. As procuras mais comuns são feitas utilizando uma lista de palavras-chave que são designadas por termos.

Information retrieval é diferente da *data retrieval* que é usada nas bases de dados utilizando consultas SQL. Os dados nestas bases de dados estão estruturados e guardados em tabelas relacionais contudo a informação no texto não é estruturada e não existe uma linguagem estruturada, como o SQL, para *information retrieval*.

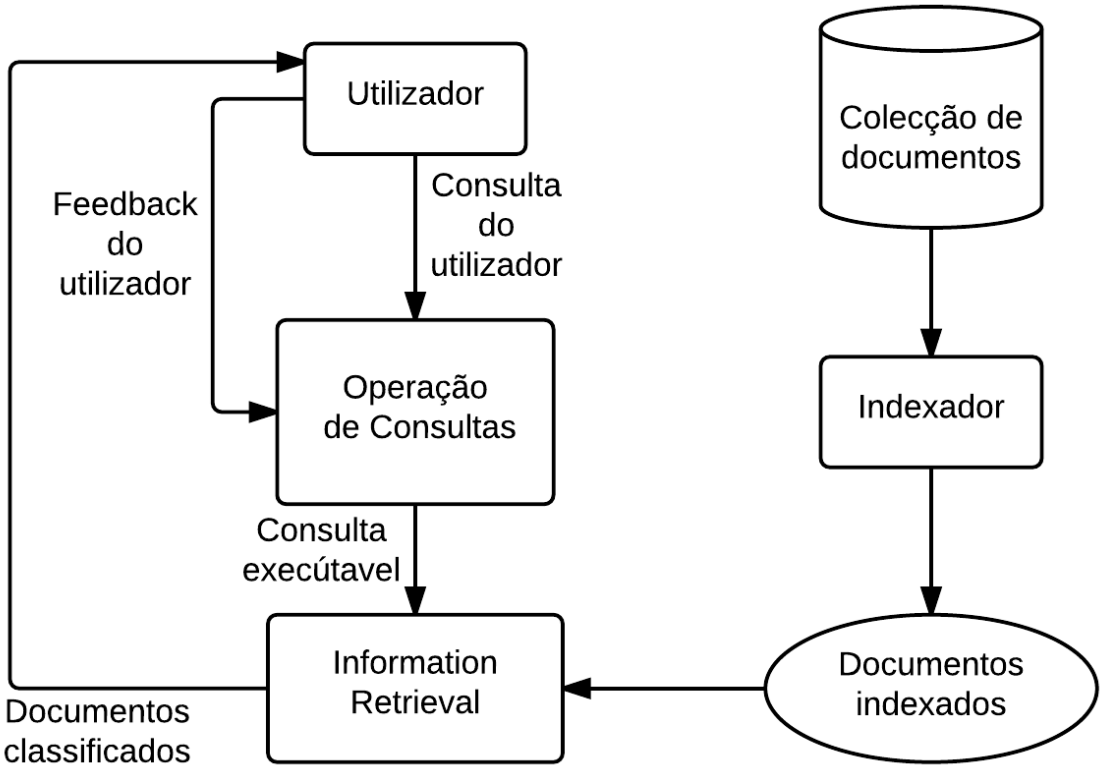

Figura 2.1: Arquitetura de um sistema de *Information Retrieval*

A Figura 2.1 mostra o processo que é feito quando o utilizador faz uma consulta a um sistema de *information retrieval*. Esta consulta passa através de um módulo de operação de consultas que por sua vez envia uma consulta executável para o módulo de *information retrieval*.

O módulo de *information retrieval* utiliza um indexador de documentos para devolver os documentos que contêm alguns dos termos da consulta, ou seja, os que são relevantes para a consulta. De seguida o módulo de *information retrieval* calcula a pontuação que irá dar a cada um dos documentos devolvidos de acordo com a sua relevância criando um conjunto de documentos ordenados pela sua pontuação que irá ser apresentado ao utilizador. A coleção de documentos é onde estão armazenados os documentos e o módulo

indexador servirá para indexar os documentos de forma a devolvê-los de forma eficiente.

Depois do processo concluído pode existir um método no qual seja possível o utilizador dar feedback sobre o ranking de documentos de forma a melhorar a qualidade dos rankings.

2.2.1 Pré-processamento

Antes dos documentos serem usados para o sistema de *information retrieval*, são necessários alguns métodos de pré-processamento. Para os tradicionais documentos de texto é usual ser utilizada uma tarefa de remoção de *stopwords*. Liu [2007], define *stopwords* como palavras que ocorrem frequentemente num texto de uma determinada língua que ajudam a construir frases mas não são relevantes para o conteúdo dos documentos e para a consulta feita.

Alguns exemplos de *stopwords* são artigos, preposições, conjunções e pronomes. Foi criado um estudo pelo departamento de ciência da computação da Universidade de Neuchatel por Savoy [2011], que faz a recolha das *stopwords* de várias línguas diferentes. Como esta tese irá ter como caso de estudo artistas lusófonos, irá ser usada apenas a lista de *stopwords* da língua portuguesa.

2.2.2 Modelos

Um modelo de *information retrieval* controla como os documentos e as consultas são representados e como a relevância de um documento para uma consulta é definido. Existem 4 modelos principais: *Boolean model*, *vector space model*, *language model* e *probabilistic model*.

Todos os modelos tratam cada documento ou consulta como termos, ou seja, cada documento e consulta é descrito por um conjunto de termos distintos. Geralmente, os modelos mais utilizados num sistema de *information retrieval* são o *boolean model* e o *vector space model* que vão ser descritos nas subsecções seguintes.

Boolean Model

Boolean model é o modelo de *information retrieval* mais básico. Os livros Manning et al. [2008], Liu [2007], Baeza-Yates et al. [1999] que retratam *information retrieval*, referem que *boolean model* usa consultas na forma de expressão de lógica booleana e a noção de combinação exata para fazer a correspondência entre os documentos e a consulta, ou seja, os documentos são retornados se tornarem a consulta verdadeira.

Representação dos documentos: Os documentos e as consultas são representados como conjuntos de termos, ou seja, cada termo individual é considerado presente ou ausente do documento.

Representação das consultas: Nas consultas, os termos são combinados usando os operadores lógicos AND, OR e NOT. Por exemplo, se a consulta for (x OR y) AND (NOT z), significa que os documentos devolvidos irão conter x ou y mas não contêm z.

Devolução dos documentos: dada uma consulta, o sistema retorna todos os documentos que fazem a consulta logicamente verdadeira. A devolução dos documentos é baseada em decisão binária, ou seja, um documento ou satisfaz ou não a consulta, isto é chamado de combinação exata.

A maior desvantagem deste modelo é não haver a noção de correspondência parcial ou ranking dos documentos retornados, o que por vezes pode levar a resultados pobres. Portanto a frequência que os termos ocorrem num documento e a proximidade entre eles, contribui significativamente para calcular a relevância de um documento. Por esta razão, na prática, é raro ser usado este modelo em separado.

Vector Space Model

Segundo Liu [2007], este é o modelo mais utilizado em *information retrieval*. Manning et al. [2008] definem *vector space model* como um modelo no qual as consultas são texto livre. Isto significa que, em vez de usar algumas palavras como no caso do *boolean model*, é usado texto separado com operadores que constroem uma expressão, que por sua vez é usada em forma de consulta. Com esta consulta o sistema decide quais os documentos que a satisfazem.

Representação dos documentos: Um documento é representado através de um vetor no qual cada componente é um termo da consulta. O peso é calculado de acordo com o esquema TF-IDF. Em TF (*Term Frequency*) o peso de um termo num documento é o número de vezes que este termo aparece no documento.

$$tf_{td} = \frac{f_{td}}{max\{f_{1d}, f_{2d}, ..., f_{[V]d}\}} \tag{2.1}$$

A fórmula 2.1 representa uma das maneiras de calcular o TF na qual o t representa os termos e o d o documento.

Contudo, este esquema não considera uma situação em que o termo apareça em vários documentos da coleção, pelo que será necessária a inclusão do IDF (*inverse document frequency*) de forma a considerar este caso.

$$idf_t = \log \frac{N}{df_t} \tag{2.2}$$

A fórmula 2.2 representa a forma de calcular o IDF na qual o N representa todos os documentos do sistema e df_t o número de documento no qual o termo t aparece pelo menos uma vez.

$$tfidf_{td} = tf_{td} \times idf_t \tag{2.3}$$

De forma a calcular o TF-IDF conjunto apenas basta multiplicar as duas equações anteriores como mostra a fórmula 2.3.

TF-IDF é o sistema de cálculo de peso dos termos nos documentos mais famoso. Existem várias formas de o calcular dependendo da *framework* utilizada.

Representação das consultas: A consulta é representada exatamente da mesma forma do que os documentos. O peso de cada termo pode também ser computado da mesma maneira.

Devolução dos documentos: Ao contrário do *boolean model*, o *vetor space model* não toma uma decisão se um documento é relevante ou não. Os documentos são classificados de acordo com os seus graus de relevância para calcular a sua similaridade com a consulta.

A similaridade com a consulta é calculada através dos mesmos vetores utilizados para o TF-IDF. Contudo, se considerarmos apenas a comparação direta de dois vetores de documentos diferentes, estes podem ser diferentes mesmo que o seu conteúdo seja semelhante. Este fenómeno pode dever-se a um dos documentos ser maior que o outro ou, por exemplo, frequência dos termos de um dos documentos ser maior que no outro. De forma a contornar este problema é usado o *cosine-similarity*.

$$cosine(d_j, q) = \frac{\langle d_j \cdot q \rangle}{\|\mathbf{d_j}\| \times \|\mathbf{q}\|} = \frac{\sum_{i=1}^{\{V\}} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{\{V\}} w_{ij}^2} \times \sqrt{\sum_{i=1}^{\{V\}} w_{iq}^2}} \tag{2.4}$$

A fórmula 2.4 representa o cálculo do *cosine-similarity* no qual d_j representa o documento e q a interrogação.

2.2.3 Indexação

O método básico de *information retrieval* é encontrar documentos que contêm os termos que estão presentes na consulta efetuada. Dada uma consulta são verificados os documentos que se encontram coleção de documentos de uma forma sequencial para encontrar aqueles que contêm os termos da consulta. Contudo isto é impraticável para uma grande coleção de dados visto que o tempo necessário para o retorno dos resultados iria ser proporcional à quantidade de dados contidos na coleção.

De forma a tornar praticável a procura na coleção de documentos é necessária a construção de estruturas de dados (índices) de forma a tornar mais rápida a pesquisa e o retorno dos documentos. Segundo Baeza-Yates et al. [1999], de todos os métodos de indexação existentes o que se mostrou superior foi o índice invertido.

O índice invertido de uma coleção de documentos é basicamente uma estrutura de dados que atribui a cada termo distinto a uma lista de todos documentos que contêm o termo.

Assim para o retorno dos dados é necessário tempo constante para encontrar os documentos que contêm um termo da consulta. Dado um conjunto de documentos D, cada

documento está identificado como um identificador único (ID). O índice invertido consiste em 2 partes: construção de um vocabulário V que contém todos os termos distintos do conjunto dos documentos e para cada termo distinto uma lista de índices invertidos. Cada elemento desta lista de índices invertidos guarda o ID do documento que contém o termo e outras informações relevantes sobre os termos dentro do documento. Dependendo do algoritmo de *retrieval* diferentes informações relevantes serão colocadas na lista de índices invertidos. Os elementos da lista de índices invertidos são ordenados por ordem crescente baseados nos IDs. Isto facilita a compressão do índice invertido.

Baseado no exemplo bastante esclarecedor de Liu [2007], o seguinte exemplo apresenta o funcionamento de índices invertidos.

Tendo três documentos id_1, id_2 e id_3:

id_1:	A	indexação	torna	as	pesquisas	rápidas		
	1	2	3	4	5	6		
id_2:	A	indexação	é	feita	através	de	índices	invertidos
	1	2	3	4	5	6	7	8
id_3:	Foi	criada	uma	lista	de	índices	invertidos	
	1	2	3	4	5	6	7	

Os números em baixo de cada documento representam o ID de cada termo em cada documento. O vocabulário é o conjunto:

{Indexação, torna, pesquisas, rápidas, feita, através, índices, invertidos, foi, criada, lista}

Segundo a lista de *stopwords* referidas na Secção 2.2.1, as palavras "a" , "as", "é", "de" e "uma" serão removidas. Depois do pré-processamento do texto são gerados os índices invertidos.

Indexação:	id_1, id_2	Indexação:	<id_1,1,[2] >, <id_2,1,[2]>
Torna:	id_1	Torna:	<id_1,1,[3]>
Pesquisas:	id_5	Pesquisas:	<id_1,1,[5]>
Rápidas:	id_1	Rápidas:	<id_1,1,[6]>
Feita:	id_2	Feita:	<id_2,1,[4]>
Através:	id_2	Através:	<id_2,1,[5]>
Índices	id_2 , id_3	Índices:	<id_2,1,[7]>, <id_3,1,[6]>
Invertidos:	id_2, id_3	Invertidos:	<id_2,1,[8]>, <id_3,1,[7]>
Foi:	id_1	Foi:	<id_3,1,[1] >
Criada:	id_2	Criada:	<id_3,1,[2] >
Lista:	id_3	Lista:	<id_3,1,[4] >

A tabela da esquerda é uma tabela de índices invertidos simples que apenas mostra cada termo e o seu documento correspondente. A tabela da direita mostra um índice

invertido do mesmo exemplo mas mais complexa em que para além do documento correspondente, mostra também o número de vezes que se encontra no documento e o ID dentro do documento.

2.2.4 Avaliação

Para avaliar a eficácia de um sistema de *information retrieval*, ou seja, a qualidade dos seus resultados, é necessária a coleção dos documentos devolvidos e a consulta efetuada. Geralmente são utilizadas duas medidas bastante populares: *precision* e *recall*.

$$Precision = \frac{\#(documentos\ relevantes\ obtidos)}{\#(documentos\ obtidos)} \tag{2.5}$$

$$Recall = \frac{\#(documentos\ relevantes\ obtidos)}{\#(documentos\ relevantes)} \tag{2.6}$$

Manning et al. [2008] define *precision* como a fração dos documentos obtidos que são relevantes e *recall* a fração dos documentos relevantes que são devolvidos, traduzindo para equação, obtemos as fórmulas 2.5 e 2.6.

Estas fórmulas podem ser representadas de outra forma, recorrendo às noções:

- *True positive* – número de classificações corretas de exemplos positivos
- *True negative* – número de classificações corretas de exemplos negativos
- *False positive* – número de classificações incorretas de exemplos positivos
- *False negative* – número de classificação incorretas de exemplos negativos

	documentos relevantes	**documentos não relevantes**
obtidos	true positive	false positive
não obtidos	false negatives	true negatives

Tabela 2.1: Matriz de confusão entre documentos obtidos e relevantes

Fazendo uma relação entre estas noções e os documentos obtidos e relevantes, obtemos a Tabela 2.1.

$$Precision = \frac{tp}{tp + fp} \tag{2.7}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.8}$$

A partir destas relações podemos chegar às fórmulas 2.7 e 2.8.

Liu [2007] apresenta um exemplo que põe em prática estas medidas. Um conjunto de dados de teste tem 100 exemplos positivos e 1000 exemplos negativos. Depois de uma

consulta feita por estes dados a um sistema de *information retrieval*, dos 100 exemplos positivos apenas considerou 1 positivo e os outros 99 negativos e dos 1000 exemplos negativos considerou todos negativos. Aplicando as fórmulas anteriores conseguimos facilmente dizer que este sistema tem 100% de *precision* e apenas 1% de *recall*.

2.3 Projetos e ferramentas relacionadas

2.3.1 "SocialBus"

As redes sociais oferecem-nos uma *API*[1] que nos permite ter acesso ao que é partilhado publicamente pelos seus utilizadores, contudo o acesso a esta informação está muitas vezes limitado a um certo número de pedidos às APIs durante um certo período de tempo imposto pelas redes sociais. Para este tipo de projetos é essencial o acesso aos dados produzidos anteriormente e ao longo do tempo. Assim sendo, o uso direto das *APIs* pode não ser suficiente.

O "SocialBus" é uma continuação do projeto TwitterEcho descrito por Boanjak et al. [2012], Oliveira [2010] e na Secção 2.1.4. O TwitterEcho é um sistema de recolha e análise de dados nas redes sociais que usa as *APIs* do *Twitter* para recolher a informação que é partilhada nestas duas redes sociais, em tempo real. O "SocialBus" adicionou também a recolha aos dados da rede social Facebook.

Esta ferramenta visa resolver algumas das limitações impostas pelas *APIs*, disponibilizando um grande conjunto de funções que permite que os utilizadores pesquisem nesta grande base de dados de informação.

O "SocialBus" funciona através de *crawlers* que estão constantemente a recolher o que é partilhado nas duas redes sociais. Estes *crawlers* são denominados consumidores visto que consumem os dados das redes sociais, guardando-os.

Existem portanto dois consumidores:

1. O consumidor do *Twitter* é usado para monitorizar e recolher as mensagens trocadas na rede social *Twitter*, que são chamadas *tweets*. Os utilizadores podem definir quais os termos de pesquisa sobre os quais são devolvidos todos os *tweets* que refiram esses termos. É também possível recolher todos os *tweets* feitos por determinados utilizadores ao longo do tempo.

2. O consumidor do *Facebook* é bastante parecido com o do *Twitter* com a diferença que apenas recolhe os *posts* públicos, ao contrário do *twitter* que consegue recolher todos porque todos os *tweets* são públicos.

[1]Link para a API do Twitter: `https://dev.twitter.com/`
Link para a API do Facebook: `https://developers.facebook.com/`
[2]Link para o "SocialBus": `http://reaction.fe.up.pt/socialbus/`

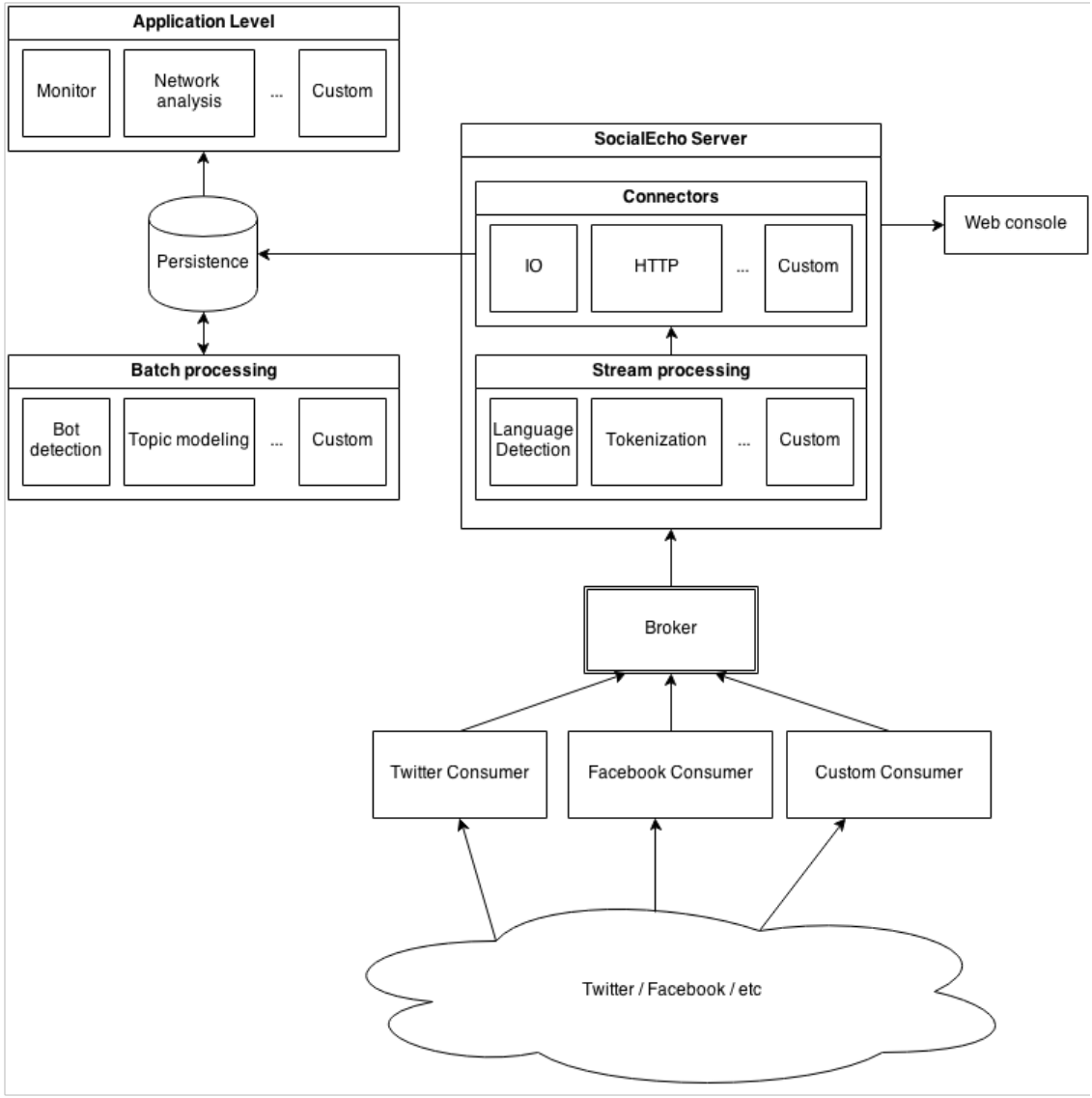

Figura 2.2: Arquitetura do "SocialBus" [2]

Quando um dos consumidores recebe um *tweet* ou *post*, envia a mensagem para o servidor do "SocialBus" que é responsável por processamento, extração de meta dados, indexação, *tokenizing* e outros cálculos como podemos ver na Figura 2.2.

2.3.2 Lucene

A ferramenta Apache Lucene[3] é um *software open-source* de procura através de uma *API* de indexação de documentos, escrito na linguagem de programação Java e desenvolvido pela Apache Software Foundation.

[3]Link para o Lucene: `http://lucene.apache.org/core/`

Segundo Gospodnetic et al. [2005], esta ferramenta funciona através da indexação de documentos, *parsers* de informação e *queries* para consultar e devolver a informação indexada de forma rápida, eficiente e com resultados fiáveis, ou seja, é uma ferramenta de *information retrieval* que funciona como o descrito na Secção 2.2.

Esta ferramenta vai ser usada neste projeto para construir o sistema de mapeamento das mensagens partilhadas nas redes sociais com a obra respetiva. Foi escolhida não apenas com base na sua eficiência e fiabilidade, como também pelas suas características de implementação. Gospodnetic et al. [2005] aponta as principais caraterísticas desta ferramenta:

- Procura baseada em rankings: os melhores resultados são apresentados em primeiro lugar.
- Vários tipos de consultas: isto permite uma grande variedade de pesquisas à informação indexada, por exemplo, TermQuery que permite fazer consultas que façam correspondência direta com os termos dos documentos e BooleanQuery que permite fazer consultas que façam correspondência de combinações booleanas de consultas.
- Procura por campos: As procuras podem ser feitas separadamente por cada campo ou todos os campos agrupados como um todo.
- Operadores booleanos: São utilizados os operadores booleanos (AND, OR, NOT) de forma a fazer combinações entre os termos de procura.
- Permite procura e atualização da informação em simultâneo.

Relembrando a Figura 2.1, o sistema de *information retrieval* é dividido em duas fases: indexação dos documentos e a procura aos documentos indexados. O Apache Lucene oferece-nos uma API em Java de forma a satisfazer estes dois objetivos. Para a indexação estas são as principais classes:

- IndexWriter: que cria um novo índice e adiciona os documentos a um índex já existente.
- Directory: representa a localização de um índex no Apache Lucene.
- Analyser: Extrai partes do texto para ser indexado e elimina o resto.
- Document: Representa uma coleção de campos.
- Field: Representa uma parte dos dados que pode ser consultado ou devolvido da indexação durante a procura.

Para a procura são necessárias as seguintes classes:

- IndexSearcher: cria um novo índice para os dados que vão ser comparados com os documentos indexados.

- Query: Representa a consulta que irá ser feita e que poderá ter vários subtipos de acordo com a pesquisa necessária.

- Hits: Conjunto de apontadores para os dados resultantes da procura.

Como referido na Secção 2.2.2, existem vários modelos de *information retrieval* para calcular a relevância dos documentos em relação à consulta.

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \in q}(tf(t \in d) \times idf(t)^2 \times t.getBoost()) \tag{2.9}$$

O Apache Lucene faz este cálculo recorrendo à classe Similarity que combina o *boolean model* 2.2.2 com *vector space model* 2.2.2 e ainda adiciona medidas de normalização. A fórmula final do cálculo é a apresentada na Fórmula 2.9.

Para o cálculo desta função é usada uma *query*(q) e um documento(d), a partir dos quais são calculados:

- coord: fator baseado no número de termos da *query* que são encontrados no documento. Um documento que contém mais termos da *query* irá ter um *score* maior do que um que tenha menos.

- queryNorm: fator de normalização que serve para encontrar um valor que consiga ser comparável entre *queries*. Por exemplo, se tivermos um documento pequeno em comparação com um grande o valor do *score* pode ser díspar, esta função faz com que consigamos ter uma unidade comparável.

- tf(*term frequency*): número de vezes que um termo aparece no documento ou seja, a ocorrência de um termo.

- idf(*inverse document frequency*): inverso do número de documentos nos quais um termo aparece, ou seja, é dado um maior peso a um documento que tenha um palavra que exista com pouca frequência em todos os documentos.

- boost: fator que permite dar um peso maior a alguma palavra no documento. Se um documento tiver um *match* com essa palavra, o *score* é multiplicado por t.getBoost(). O valor t.getBoost() por defeito é 1 para este fator não ser considerado.

2.3.3 "Music Timeline"

O "Music Timeline" é um projeto dos grupos de investigação Big Picture e Music Intelligence do Google. Este projeto mostra os géneros musicais de acordo com a sua popularidade na ferramenta Google Play Music.

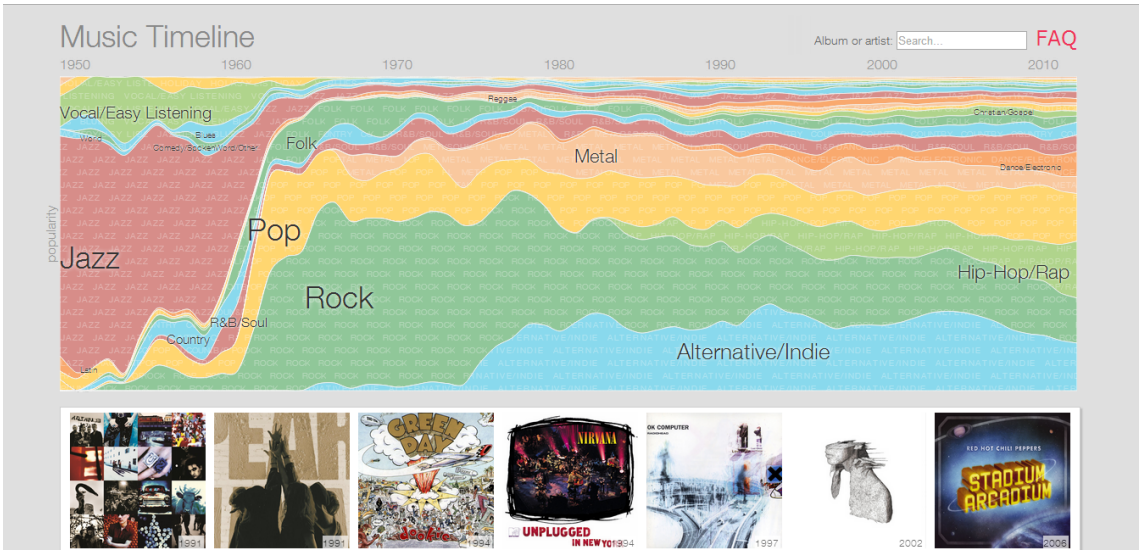

Figura 2.3: Página principal de "Music Timeline"

Como se pode verificar na Figura 2.3, cada camada no gráfico representa um género musical e a densidade do gráfico mostra a popularidade da música lançada num determinado ano desse género, por exemplo, a camada que corresponde ao género Jazz é mais densa em 1950, isto significa que existem mais utilizadores que adicionaram às suas bibliotecas no Google Play Music álbuns de jazz lançados em 1950.

Esta representação em forma de gráfico ao longo do tempo é bastante simples, intuitiva e direta. Visto que o projeto "Lusica" se trata do mesmo contexto, a representação da sua informação irá ser baseada na representação do "Music Timeline".

Capítulo 3

"Social Impact"

3.1 "O Mundo em Pessoa"

Antes da criação da arquitetura "Social Impact", foi criado um projeto que apenas estava focado no contexto da obra de Fernando Pessoa, "O Mundo em Pessoa". Este foi um projeto de recolha automática de citações de Fernando Pessoa (ortónimo e heterónimos) a partir das redes sociais mais utilizadas (Facebook e Twitter). Foi lançado em 2013, quando se comemorou os 125 anos do nascimento do poeta. Pessoa, sendo o poeta português mais conhecido dentro e fora de Portugal, é também provavelmente o mais citado.

Com este projeto foi possível identificar quais os versos e frases de Fernando Pessoa que mais inspiram os seus leitores de todo o mundo. Outro objetivo deste projeto é também conduzir todos aqueles que usam as palavras de Pessoa até ao seu texto original, ampliando o número de leitores e o conhecimento da sua obra.

Sempre que é citado um texto de Fernando Pessoa no Twitter ou em páginas públicas do Facebook, "O Mundo em Pessoa" identifica e mostra essa mensagem numa interface própria. Para validar se um texto é uma citação da obra de Fernando Pessoa, este é comparado com arquivos da obra do poeta disponíveis *online*.

Para capturar os requisitos de "O Mundo em Pessoa" foi então construído um sistema de informação *web* orientado a serviços. Este sistema tem as três camadas habituais neste tipo de sistema: *front-end*, *web services* e *back-end*.

3.1.1 *Front-end*

O *front-end* deste projeto foi desenvolvido, com base na parceria com o SAPO Labs, por uma equipa de *designers* e *web-developers* desta empresa. Como se pode verificar na Figura 3.1, "O Mundo em Pessoa" mostra através de uma apresentação em mosaicos, as mensagens publicadas nas redes sociais, bem como a foto de perfil do utilizador que a publicou e ligações para a mensagem, perfil do utilizador e para a obra correspondente à mensagem publicada. Em cima dos mosaicos existem filtros, que permitem aos utilizadores filtrar as mensagens que aparecem nos mosaicos por heterónimos ou pelo ortónimo.

Existe também a possibilidade de navegar ao longo do tempo, neste caso podemos observar as publicações do próprio dia, da própria semana e do próprio mês. Por fim possui também a possibilidade de o utilizador saber quais são as obras mais citadas e respetivos autores (heterónimos ou ortónimo).

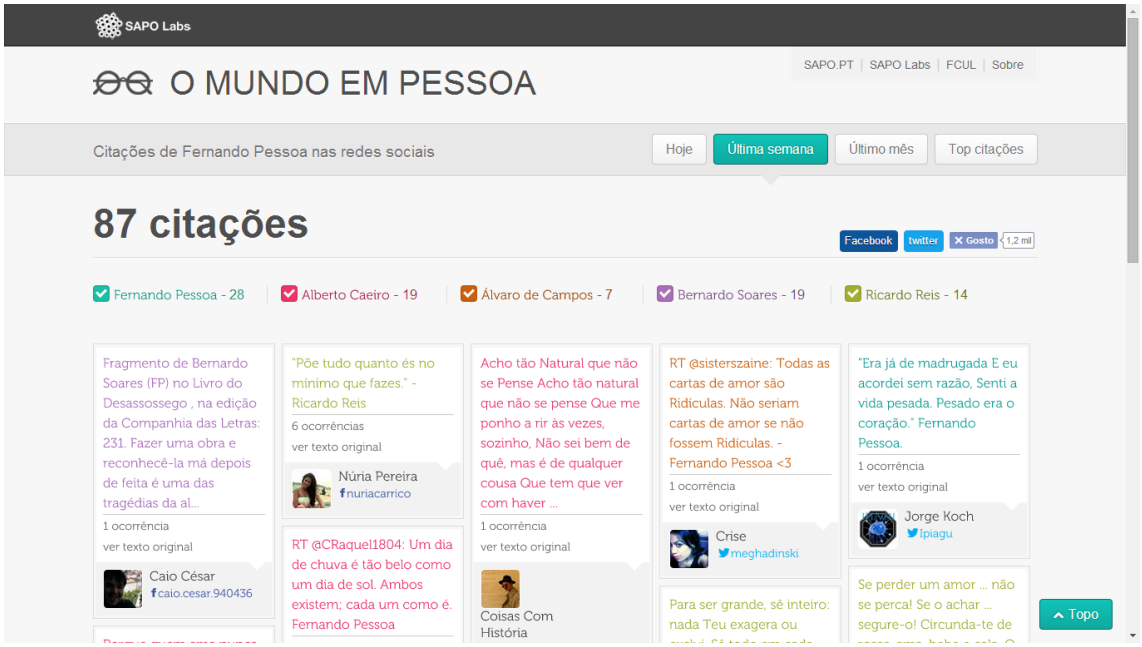

Figura 3.1: Página principal de “O Mundo em Pessoa”

3.1.2 *Web services*

Os *web services* são serviços disponibilizados que permitem fazer a integração do resto do sistema com o *front-end*. Estes serviços vão fazer interrogações à base de dados de acordo com os parâmetros passados pelo *front-end* através de um URI, funcionando assim como um *broker* entre a informação que está na base de dados e a aplicação que a vai buscar. Neste projeto, os *web services* são baseados no estilo da arquitetura REST que foram desenvolvidos na linguagem de programação PHP.

3.1.3 *Back-end*

O back-end, de forma a recolher a informação e fazer a correspondência entre as mensagens nas redes sociais e a obra de Fernando Pessoa, utiliza sistemas de *web mining* e de *information retrieval*. Está divido em 3 componentes fundamentais: base de dados, *crawlers* e *wrappers* e algoritmo de validação de citações incorretas.

Para a base de dados, foi construída uma base de dados relacional simples em MySQL de forma a guardar os dados necessários para a aplicação funcionar.

O projeto começou pela recolha da informação existente na internet. Visto que era necessária a comparação entre a obra e as citações nas redes sociais, foram necessários *crawlers* que procurassem estes dois tipos de informação e *wrappers* que recolhessem a informação de forma a ser guardada na base de dados.

Como a obra de Fernando Pessoa nunca será alterada, apenas bastou encontrar serviços fiáveis na *web* que devolvessem esta informação. Assim sendo foram retiradas de duas fontes: "Arquivo Pessoa"[1] e "Casa Pessoa"[2]. O sistema que fez esta recolha foi apenas executado uma única vez obtendo assim todas as obras disponíveis de Fernando Pessoa.

Para recolher as citações nas redes sociais foi necessário utilizar a API fornecida pelas redes sociais. De ambas as *APIs* foi utilizado o método *search* que devolve as citações mais recentes de um determinado termo. Assim sendo, foi criada uma lista com os termos que vão ser usados no método *search* e retornam as últimas mensagens que os citem. Como este método apenas devolve as últimas citações é necessário um programa que esteja constantemente a correr. No entanto, as *APIs* tem limite de pedidos por hora, por tanto, entre cada vez que o programa vai buscar as últimas citações, é feito um *sleep* de forma a esperar até ao próximo pedido.

Os programas que retiravam informação das redes sociais, tinham também a função de verificar o texto a que pertenciam. Depois dos dados recolhidos das redes sociais foi necessário um sistema para fazer a correspondência entre estes dados e as obras de Fernando Pessoa. O método que estava a ser utilizado para esta função consistia em comparações de *strings* de 3 palavras de cada vez, o texto que fizesse o maior *match* era o escolhido. Contudo, verificou-se que este não era o melhor método porque foram obtidas citações que não correspondiam realmente a obras.

Para resolver este problema, modificou-se o algoritmo de validação de citações adicionando um *script* que validava estas citações. Este *script* verifica se o texto contém palavras comuns (*stopwords*) e palavras que não correspondem de certeza a poemas de Fernando Pessoa (*badwords*). Depois disto é atribuída uma pontuação de acordo com um número de matches com a obra original. De acordo com esta pontuação é definido um *trade off* de forma a definir até que pontuação a citação será definida como válidas ou inválidas.

3.1.4 Problemas

Esta foi uma primeira abordagem construída apenas com o principal objetivo de cumprir o prazo do aniversário de Fernando Pessoa. Foi portanto utilizada uma abordagem mais ágil de forma a cumprir este objetivo. Foram portanto deixados vários pontos deste projeto como trabalho futuro:

[1]Link para o "Arquivo Pessoa": `http://arquivopessoa.net/`

[2]Link para a "Casa Pessoa"`http://casafernandopessoa.cm-lisboa.pt/index.php?id=2241`

- Necessária uma otimização da base de dados de modo às consultas serem conseguidas de forma mais eficiente através da criação de índices e chaves estrangeiras. Os dados recolhidos nem sempre são consistentes, por vezes repetidos no caso das obras e alguns problemas com *encodings*.

- Alguns serviços precisam de ser refeitos de forma às consultas na base de dados serem conseguidas de forma mais eficiente.

- O algoritmo utilizado para deteção de citações incorretas é bastante simples, baseia-se na procura direta de palavras, usando uma lista de *stopwords* e uma lista de termos caluniosos a serem eliminados. Isto traduz-se numa fiabilidade de resultados obtidos limitada. O *trade off* que foi usado para considerar uma mensagem citação, fazia com que algumas citações das redes sociais que realmente correspondiam a um texto do Fernando Pessoa fossem consideradas incorretas e algumas citações que não correspondiam a textos de Fernando Pessoa fossem consideradas como tal.

3.2 Requisitos

O principal objetivo desta tese, como descrito na Secção 1.2, é construir um sistema que faz o mapeamento de uma mensagem partilhada nas redes sociais com a obra que está a ser citada. Para este fim, foi proposto um sistema para “O Mundo em Pessoa”, descrito em 3.1.3, que passava pela comparação direta de *strings* e que não oferecia grande fiabilidade de resultados. Para além deste principal problema e dos problemas detetados e descritos na Secção 3.1.4, havia a necessidade de criar um sistema abstrato que não esteja preso a um contexto mas que seja extensível a vários.

Tendo em conta todos estes problemas, em seguida serão apresentados os requisitos funcionais e não funcionais para a arquitetura “Social Impact” que resolverá todas estas questões.

Requisitos não funcionais

Os requisitos não funcionais referem-se a aspetos estruturais do sistema para que seja garantido que o sistema funciona. No caso desta arquitetura irão ser os *inputs* necessários que irão depender do contexto a que o sistema irá ser submetido. Então os requisitos não funcionais desta arquitetura serão:

1. Obra artística: A procura pela informação das obras artísticas de um determinado autor será feita de forma manual visto que os contextos podem ser completamente distintos como por exemplo, obras literárias e obras musicais.

   Para a recolha da informação será necessário recorrer a métodos de *web data mining* como *crawlers*, descritos na Secção 2.1.2, de forma a percorrer as fontes selecio-

nadas manualmente e *wrappers* como os descritos na Secção 2.1.3. Será também necessário ter em conta a integração da informação como descrito na Secção 2.1.3 de forma a não conter dados inconsistentes ou repetidos na base de dados como aconteceu no primeiro protótipo de "O Mundo em Pessoa".

2. Lista de termos associados às obras: esta lista irá conter os termos que serão procurados nas redes sociais de forma a encontrar citações partilhadas pelos seus utilizadores. Esta lista será única para cada contexto e poderá conter nomes dos autores das obras, título da obra, etc. Por exemplo, esta lista poderá ser, se considerarmos o exemplo de uma obra literária, o nome do autor, nomes dos heterónimos ou pseudónimos e o nome das obras mais famosas.

Requisitos funcionais

Os requisitos funcionais descrevem as funcionalidades que se espera que o sistema disponibilize, atendendo aos propósitos para qual o sistema será desenvolvido. As principais funcionalidades desta arquitetura serão:

1. Guardar e organizar informação: será necessário um sistema que guarde a informação que é dada como *input* e esteja disponível de forma a que o sistema possa trabalhar sobre esta informação.

2. Recolha de mensagens nas redes sociais: é necessária a recolha automática e constante ao longo do tempo, de mensagens nas redes sociais que contenham pelo menos um termo da lista descrita no ponto 2 da lista apresentada na Secção 3.2.

3. Mapeamento das mensagens com obra: será este o principal objetivo deste sistema. Irá ser necessário o recurso a um sistema de *information retrieval* como descrito na Secção 2.2.

4. Disponibilização da informação: a informação que está guardada no sistema terá que ser disponibilizada para ser utilizada. Este processo será feito com o recurso a serviços de forma à informação ser disponibilizada de forma segura, eficiente e atualizada.

3.3 Arquitetura

"Social Impact" é uma arquitetura construída para cumprir o principal objetivo desta tese, descrito na Secção 1.2, através da implementação dos requisitos descritos na Secção 3.2. Esta arquitetura será baseada na arquitetura utilizada para o projeto "O Mundo em Pessoa" descrito na Secção 3.1 com os melhoramentos necessários em relação à última versão detetados na Secção 3.1.4.

3.3.1 Especificação

A arquitetura construída foi baseada numa estrutura em serviços. Esta estrutura faz com que seja mais fácil integrar novos componentes e assim é possível construir um sistema abstrato que possa ser estendido para vários contextos. Para além disto, a mesma estrutura foi já utilizada no projeto “O Mundo em Pessoa” descrito na Secção 3.1 o que faz com que alguns componentes possam ser reutilizados.

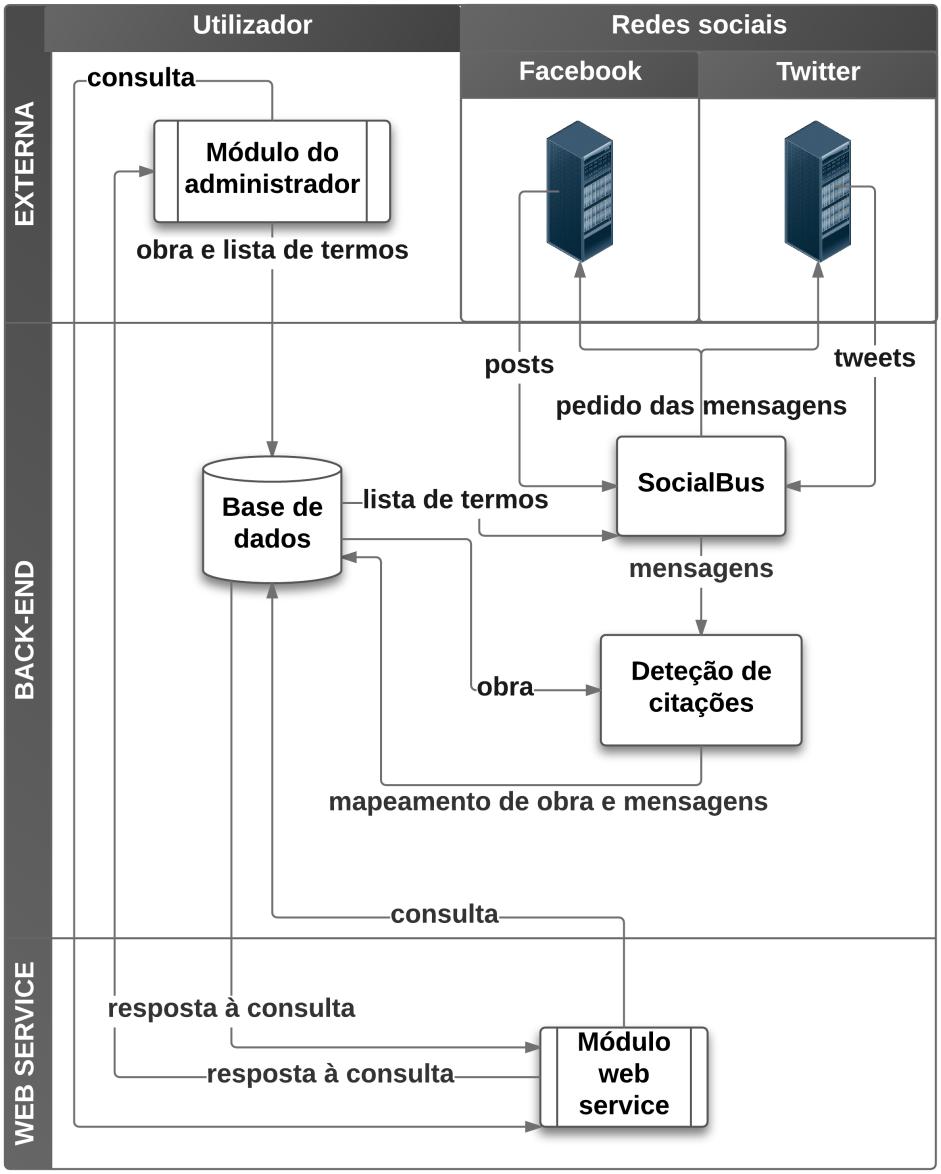

Figura 3.2: Arquitetura “Social Impact”

Este sistema está representado na Figura 3.2 e terá três camadas principais:

- Externa: camada que está ligada com o sistema mas não o integra. Está dividido em dois módulos fundamentais:

  - Módulo do administrador: faz a ligação com a base de dados fornecendo os *inputs* como requisitos do sistema. Este módulo pode ser implementado de forma ao administrador do sistema estender o sistema para vários contextos.
  - Redes sociais: neste módulo estão os serviços disponibilizados pelos servidores das redes sociais Facebook e Twitter.

- *Back-end*: camada onde é feita a recolha, processamento e armazenamento dos dados. Este é o componente principal visto que é por onde passa todo o processamento dos dados. Dentro deste componente existem 3 módulos:
  - Base de dados: onde são guardados e fornecidos todos os dados do sistema.
  - "SocialBus": faz a recolha das mensagens que são partilhadas nas redes sociais a partir da lista de termos passada do módulo do administrador.
  - Deteção de citações: dadas as obras e as mensagens partilhadas nas redes sociais, usa um sistema de *information retrieval* para fazer o mapeamento destes dois tipos de dados.

- *Web service*: Camada que consulta os dados contidos na base de dados e devolve-os de forma a serem consultados.

3.3.2 Implementação

Nível externo

Esta camada irá conter todos os componentes externos ao sistema, ou seja, todos os serviços externos utilizados pelo sistema.

Como representado na Figura 3.2, esta camada contém o componente do utilizador, que por sua vez contém o módulo do administrador e o das redes sociais no qual estão contidos os serviços disponibilizados pelas redes sociais Facebook e Twitter.

O módulo do administrador é responsável por enviar o *input* dos dados necessários e é também o que permite a extensão deste sistema de forma a ser aplicado a vários contextos, ou seja, é o módulo que permite ao administrador do sistema que irá estender esta arquitetura, adicionar ou modificar os componentes.

Como descrito na Secção 3.2, o input será a obra que está em formato de texto e uma lista, também em formato texto, com os termos a serem procurados nas redes sociais. Estes *inputs* são adicionados à base de dados que se encontra dentro da camada *back-end* para os dados serem processados.

Depois dos dados processados pelo *back-end*, o *output* resultante da computação do sistema irá estar guardado na base de dados e poderá ser consultado por este módulo através da camada *web service*. De forma a comunicar com esta camada, são feitas consultas através de um URI e o *web service* irá retornar a resposta a esta consulta, de acordo com os dados recebidos da base de dados, no formato JSON.

Estas informações podem ser utilizadas posteriormente para, por exemplo, construir um *front-end* para as mesmas tenha outra representação visual.

Back-end

A camada de *back-end* é responsável por todo o trabalho de recolha, processamento e armazenamento dos dados.

Como se pode ver na Figura 3.2, esta camada contém três componentes principais: base de dados, “SocialBus” e deteção de citações que irão ser descritos em detalhe nas próximas subsecções.

Base de dados

Para todos os dados recolhidos, analisados e validados, é necessário um sistema de armazenamento que esteja preparado para ser consultado e atualizado constantemente. O sistema utilizado irá ser implementado através de uma base de dados MySQL onde se pode armazenar informação em forma de tabelas relacionais. Esta implementação foi feita de uma forma genérica de forma a ser reutilizada para qualquer tipo de obra.

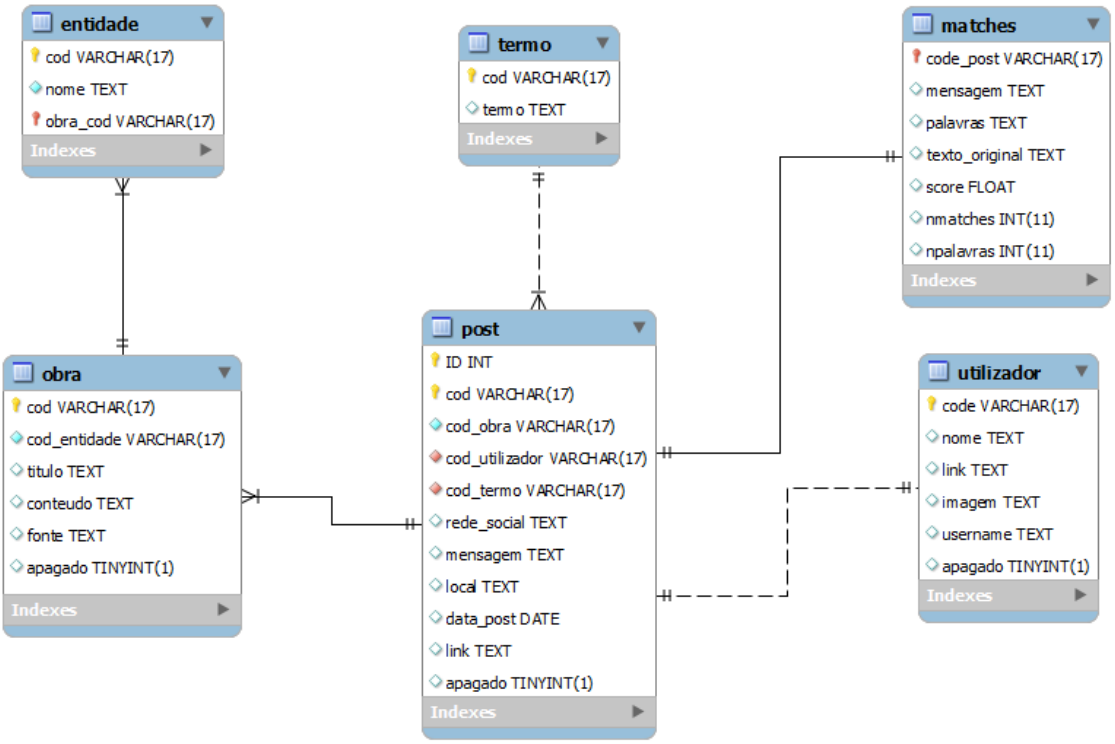

Figura 3.3: Modelo entidade associação retirado da ferramenta MySQL Workbench[3]

[3]Link para MySQL Workbench: `http://www.mysql.com/products/workbench/`

A Figura 3.3 mostra a base de dados construída para esta arquitetura. Serão portanto criadas cinco tabelas:

- Entidade: tabela que guarda informação sobre a entidade autora da obra artística. Esta tabela tem uma ligação à tabela obra de forma a relacionar as entidades a cada obra.

- Obra: guarda as informações sobre a obra que são fornecidas pelo módulo de administrador. Esta tabela é relacionada com a tabela post.

- Post: tabela central da base de dados. Esta tabela armazena todas as informações sobre as mensagens partilhadas nas redes sociais. Os campos principais desta tabela são o "cod_obra" e o "apagado". O "cod_obra" é o campo que relaciona as mensagens das redes sociais à obra e por isso por defeito tem o valor NULL, ou seja, não tem relação com a obra. O campo "apagado" serve para dizer se uma determinada mensagem deve ser retornada ou não. Esta mensagem não é mesmo apagada da base de dados para podermos fazer uma avaliação das mensagens apagadas.

- utilizador: tabela que contém a informação do utilizador das redes sociais que partilhou uma determinada mensagem.

- matches: tabela que mostra estatísticas sobre o mapeamento feito das mensagens contidas na tabela post com a obra.

"SocialBus"

Tendo a base de dados preparada para receber informação e com a informação das obras e a lista do que procurar nas redes sociais, falta a recolha das mensagens partilhadas nestas redes sociais.

Para esta análise do que é partilhado nas redes sociais, é necessário um método de recolha que nos permita obter as mensagens partilhadas de acordo com a lista de termos de pesquisa introduzido pelo módulo do administrador.

De forma a completar este objetivo é utilizado o "SocialBus" descrito na Secção 2.3.1. Em primeiro lugar, o "SocialBus" necessita de uma chave de acesso às APIs das redes sociais Twitter e Facebook. Esta chave contém vários *tokens* que são necessários para cada função da rede social e é obtida através das APIs das respetivas redes sociais. Estes *tokens* terão de ser fornecido ao "SocialBus" através de um ficheiro CSV(Comma-separated values) com o formato: <token >,<token-secret >, <consumer-key >, <consumer-secret >.

Outro requisito do "SocialBus" é a lista dos termos de procura que é passado pelo módulo do administrador. Com o *input* dado, o "SocialBus" começa a recolha de dados. Cada vez que um utilizador referir nas suas mensagens escritas um dos termos que está na

lista de termos, esta mensagem é devolvida assim como todas as informações sobre ela: data, quem partilhou, etc.

Estas informações são automaticamente guardadas num ficheiro temporário. Este ficheiro é criado dentro de uma árvore de pastas com três níveis onde o primeiro corresponde ao ano, o segundo ao mês e o terceiro ao dia. O nome do ficheiro corresponderá à hora que a mensagem foi partilhada. Este sistema permite aos utilizadores do “SocialBus” saberem facilmente onde se encontram as informações correspondentes a cada data e hora.

Como referido na Secção 2.3.1, cada rede social tem o seu consumidor para recolher as mensagens partilhadas pelos seus utilizadores, funcionando assim como dois programas independentes que resultam na geração das suas próprias pastas e ficheiros.

Detecção de citações

O sistema de deteção de citações é o componente central desta tese. Este sistema é construído utilizando a ferramenta Apache Lucene descrita na Secção 2.3.2.

O objetivo desta componente é, para cada mensagem recolhida pelo “SocialBus”, atribuir uma obra das que foram fornecidas pelo administrador do sistema e guardadas na base de dados.

Como descrito na Secção 3.3.2, o “SocialBus” guarda as mensagens partilhadas nas redes sociais em ficheiros, ou seja, para recolher estas mensagens das duas redes sociais (Twitter e Facebook), serão necessários dois ficheiros separados.

Como estes ficheiros são atualizados independentemente, para que sejam processados o mais rapidamente possível para estarem disponíveis para consulta, é necessário um sistema concorrente. Assim são criadas duas *threads* em que cada uma irá consultar um ficheiro correspondente a uma rede social e processar as mensagens de cada ficheiro e guardá-las na base de dados.

A Figura 3.4 mostra uma visão detalhada deste sistema e a relação entre este componente e os outros componentes do *back-end*. Este sistema está dividido em duas fases. Na primeira, o objetivo é fazer a indexação das obras. Na segunda, realizar a correspondência com as mensagens das redes sociais.

Em primeiro lugar o sistema faz a adição dos documentos. Os documentos neste sistema vão ser a representação da obra que está guardada na base de dados. Então o sistema faz uma consulta SQL à base de dados de forma a recolher todas as obras e adiciona o seu conteúdo e o código correspondente a uma classe Documents do Lucene.

O segundo passo será a filtragem destes documentos pelas *stopwords* como referido na Secção 2.2.1. Esta lista estará num ficheiro de texto que irá ser lido pelo sistema de forma a poder ser alterado pelo administrador para poder adicionar mais *stopwords* de acordo com o contexto a que pretende estender.

Para fazer esta verificação de *stopwords* o sistema precisa de dividir cada texto em

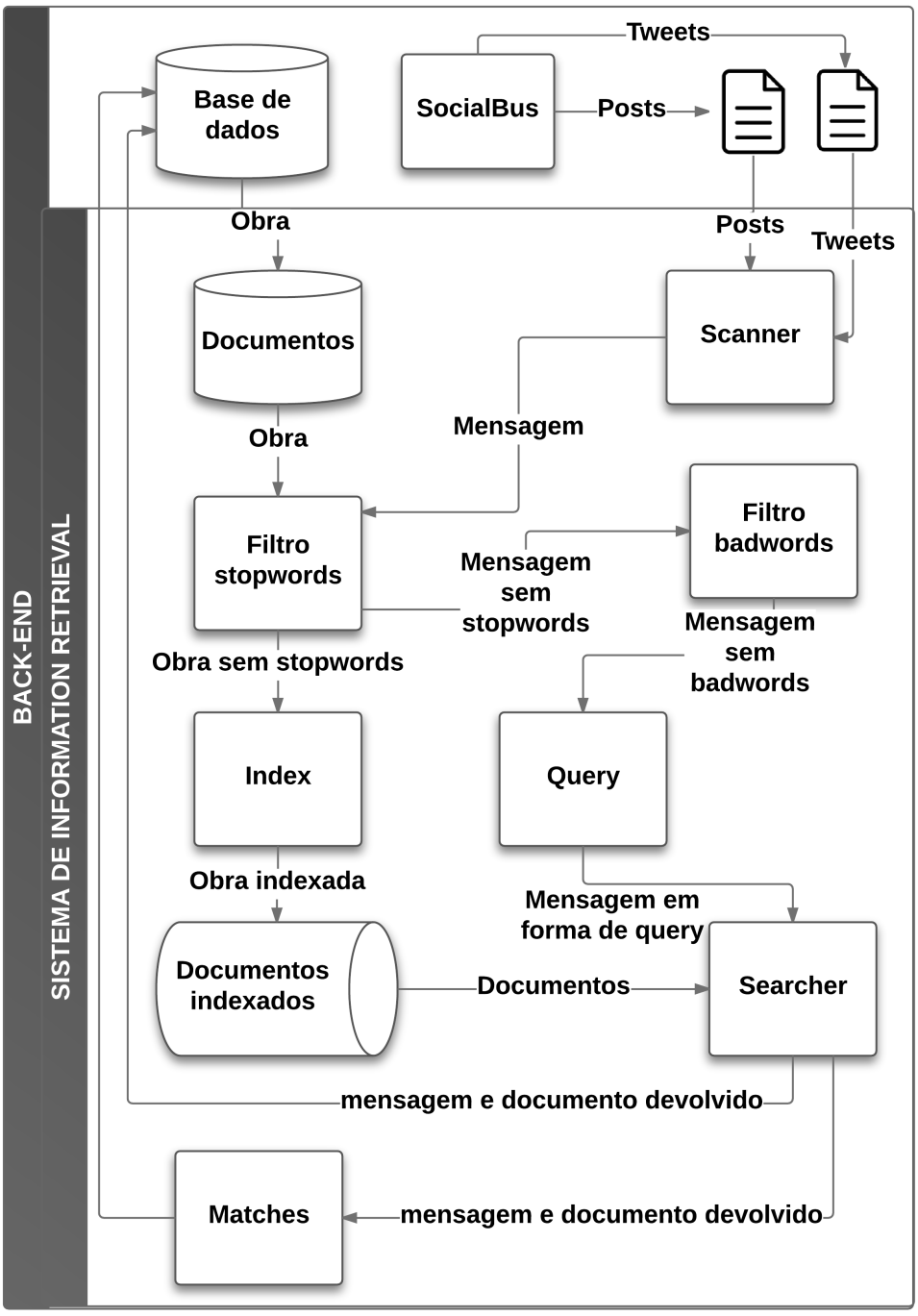

Figura 3.4: Esquema detalhado do sistema de deteção de citações

termos que neste caso irão ser as palavras do texto da obra e posteriormente, estes termos serão passados a um método de indexação.

A indexação dos documentos como descrito na Secção 2.2.3, serve para fazer pesquisas em grandes quantidades de informação de forma eficiente. O método de indexação vai receber os termos separados de cada obra e vai atribuir-lhes IDs de forma a identificar a que obra pertencem.

Depois de concluída a indexação os documentos ficam disponíveis para pesquisa e assim fica concluída a primeira fase deste sistema. Esta primeira fase apenas necessita ser executada na primeira execução do sistema caso a obra não seja alterada e assim, os documentos ficarão sempre indexados.

A segunda fase do sistema irá estar constantemente a ser executada uma vez que estará dependente das mensagens que o "SocialBus" recolher. Serão criadas então as duas *threads*, uma para o ficheiro gerado do Facebook e outra para o do Twitter. Quando estes ficheiros forem lidos, serão guardados os respetivos tamanhos, de forma a possibilitar a verificação constante da existência de novos dados a serem processados.

O texto desta mensagem, à semelhança do que aconteceu com a obra, é passado pelo filtro de *stopwords* de forma a não considerar as palavras comuns. Para além deste filtro será necessário outro método de pré-processamento, o filtro de *badwords*. Este filtro foi criado com o intuito de remover os termos caluniosos que muitas mensagens partilhadas continham. Optou-se por deixar ao critério do administrador do sistema a decisão sobre o que fazer quando um termo calunioso aparece: não considerar a mensagem uma citação, ou censurar a parte que contém um destes termos. O método de construção deste filtro é igual ao filtro de *stopwords*, contendo um ficheiro de texto com os termos caluniosos definidos pelo administrador.

Depois de a mensagem ter passado pelos filtros, é feita uma consulta com a mensagem. Esta consulta, à semelhança dos documentos, parte o texto da mensagem em termos e separa-os em operadores lógicos AND como descrito na Secção 2.2.2.

Esta mensagem em forma de consulta é posteriormente passada ao método *searcher* da ferramenta Apache Lucene que vai buscar os documentos indexados na primeira fase deste sistema e para cada documento, vai calcular o score de semelhança com a mensagem em forma de *query*, através da fórmula 2.9 descrita na Secção 2.3.2.

Deste cálculo é devolvida uma lista com o *ranking* dos documentos semelhantes ordenados pelo score calculado. O documento que tiver o maior score é mapeado à mensagem que é adicionada na tabela post da base de dados. O campo "cod_obra" será o campo que irá servir de ligação entre o post e a obra. Então este campo será preenchido com o código do documento com maior score que foi atribuído na indexação dos documentos.

Outro campo importante desta tabela é o "apagado". Este campo será um campo booleano que será considerado verdadeiro se o score calculado do documento mapeado à mensagem for maior ou igual a um valor *threshold* definido pelo administrador do sistema. Este campo ser considerado verdadeiro significará que a mensagem será considerada como citação da respetiva obra mapeada.

Este *threshold* é um *float* que irá definir a qualidade dos resultados retornados, ou seja, irá tornar os resultados retornados com mais *precision* e *recall*, como explicado na Secção 2.2.4.

Então, recordando as definições da Tabela 2.1, se o *threshold* for definido com um

valor mais baixo então haverá mais *true positives*, contudo haverá mais *true negatives* e consequentemente mais *recall* e menos *precision*. Se o *theshold* for definido com um valor mais alto haverá mais *false negatives* contudo, haverá menos *true negatives* e consequentemente mais *precision* e menos *recall*. Este valor terá que ser calculado através da avaliação dos resultados obtidos para cada contexto porque para contextos diferentes irão ser retornados valores diferentes.

Por fim, foi utilizado o método *matches* que retorna uma explicação sobre a escolha do documento considerado com sendo o citado na mensagem. Isto é feito através do retorno do valor do score e das palavras que emparelharam a obra indexada e a mensagem, que será armazenado na base de dados.

Web service

Web service é um componente utilizado na integração de sistemas e na comunicação entre aplicações diferentes.

Richardson et al. [2008] definem o grande objetivo da criação dos *web services* como a possibilidade de novas aplicações interagirem com aplicações que já existam e que, por exemplo, sistemas em plataformas diferentes sejam compatíveis visto que, independentemente da linguagem em que as aplicações são construídas, o *web service* irá traduzir os dados dos pedidos para um formato universal. Para além disto é um mecanismo mais dinâmico e seguro visto que toda a comunicação é feita através de sistemas sem intervenção humana.

Existem vários estilos de arquiteturas de *web services*. O estilo que foi utilizado nesta arquitetura foi o REST (REpresentation State Transfer) porque é um estilo flexível, ou seja, poderá optar-se pelo formato mais adequado às mensagens do sistema de acordo com uma necessidade específica. Esta vantagem será útil para esta arquitetura visto que esta será estendida para outros contextos e cada contexto irá necessitar de *web services* próprios, pelo que será necessária esta flexibilidade.

Para o acesso à informação de um *web service* REST, o pedido precisa de informar qual o seu tipo de pedido é e qual a informação pedida. Para saber o tipo de pedido são utilizados os métodos HTTP:

- GET: para receber informação
- POST: pra adicionar nova informação mostrando a sua relação com a informação antiga
- PUT: atualizar informação
- DELETE: descartar informação

Para ser feito um pedido a uma informação em específico são usados URIs (*Uniform Resource Identifier*) de forma a identificar que recurso é pretendido. Por exemplo, para

buscar uma página web, o *browser* faz um GET num URI e devolve a representação dos recursos identificados pelo URI.

Nesta arquitetura foram desenvolvidos *web services* REST na linguagem de programação PHP, visto que já existe um *background* com *web services* nesta linguagem devido ao projeto “O Mundo em Pessoa” descrito na Secção 3.1.

A resposta do *web service* virá em forma JSON(*JavaScript Object Notation*) que é um formato de dados na forma de subconjunto de objeto de *javascript*.

Este formato foi escolhido devido à sua rapidez de resposta e uso eficiente de recursos em relação à outra alternativa XML(*eXtensible Markup Language*) como se pode constatar em vários estudos de comparação com estes dois formatos: Nurseitov et al. [2009] e Wang [2011].

Operação	Consulta	Retorno
GET	obra/	retorna a informação sobre todas as obras que se encontram na base de dados
GET	citacao/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema.
GET	utilizador/	retorna a informação sobre todos os utilizadores.
GET	palavrascitadas/	retorna a informação sobre o mapeamento de todos as mensagens das redes sociais com a obra respetiva.

Tabela 3.1: Tabela dos *web services* no “Social Impact”

Na Tabela 3.1 estão definidas as operações e consultas para cada tipo de informação na base de dados e os respetivos retornos. Visto que a informação disponível na base de dados irá depender do contexto ao qual este sistema será estendido, estes *web services* terão que ser estendidos de forma a consultar a informação nova.

Também de notar que o sistema só possui operações GET visto que o *web service* deste sistema apenas está preparado para fornecer a informação sobre os dados que estão na base de dados. Conforme o contexto poderão também ser adicionados outros métodos, por exemplo, para permitir ao utilizador dar *feedback* sobre as obras retornadas de cada mensagem.

Capítulo 4

Resultados

4.1 Caso de estudo: "O Mundo em Pessoa"

Como descrito na Secção 3.1, "O Mundo em Pessoa" foi um projeto de recolha automática de citações de Fernando Pessoa (ortónimo e heterónimos) a partir das redes sociais mais utilizadas (*Facebook* e *Twitter*).

Este foi um projeto muito rudimentar apenas focado no contexto da obra de Fernando Pessoa feito para cumprir o prazo que foi o seu aniversário. De todos os problemas encontrados neste projeto, o que necessitava de uma grande melhoria era o método de detecção citações incorretas que era feito de uma forma simplista através de comparação direta de *strings*.

Foi portanto criado um segundo protótipo de "O Mundo em Pessoa" cujo objetivo será a implementação da arquitetura "Social Impact" descrita na Secção 3.3. Por outras palavras, esta nova versão de "O Mundo em Pessoa" vai ser uma extensão do sistema para o contexto das obras literárias, mais concretamente sobre a obra do poeta Fernando Pessoa.

4.1.1 Arquitetura

Este novo projeto de "O Mundo em Pessoa", permitirá que sejam resolvidos os problemas detetados na versão anterior e descritos na Secção 3.1.4. Como já se encontra construída a arquitetura "Social Impact" descrita na Secção 3.3.1, esta apenas terá que ser estendida para este contexto de modo a resolver os problemas detetados.

Relembrando a Figura 3.2, para uma extensão de contexto, é necessária a implementação do módulo de administrador, contudo existem alguns pontos em aberto nos restantes componentes que terão que ser definidos com esta extensão.

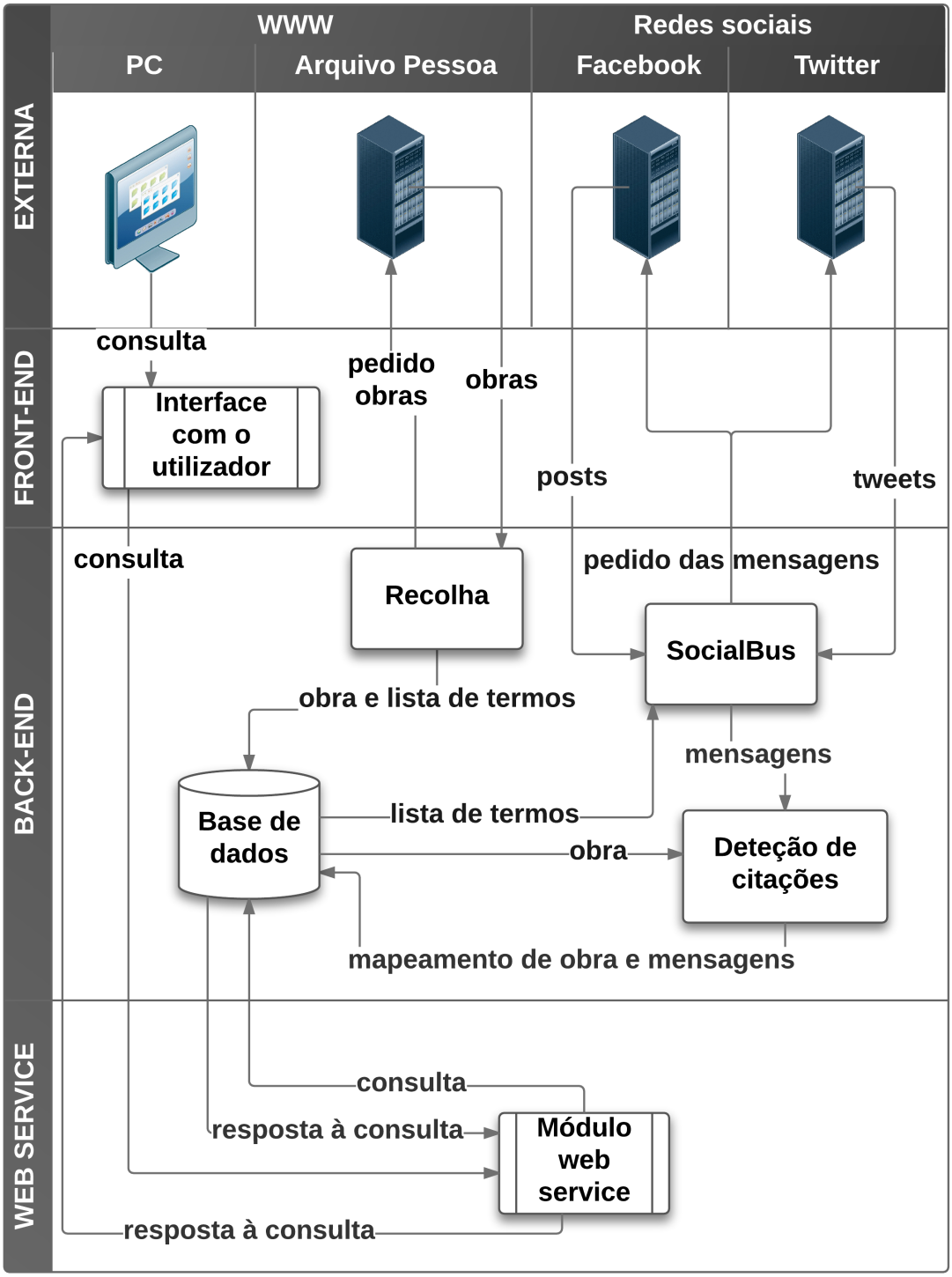

Figura 4.1: Arquitetura do sistema "O Mundo em Pessoa"

Na Figura 4.1, está representada a arquitetura construída para "O Mundo em Pessoa".

Comparando com a Figura 3.2, é possível verificar os componentes que foram estendidos para este contexto.

O componente módulo de administrador foi substituído por um sistema de recolha necessário para recolher a obra de Fernando Pessoa através do repositório Arquivo Pessoa. Foi também criada outra camada no sistema, a camada de *front-end* que contém a interface com o utilizador que faz a comunicação entre o utilizador final da aplicação e o módulo *web service* de forma a devolver a informação num formato mais legível para o utilizador comum.

Esta interface com o utilizador irá ser a mesma da primeira versão de "O Mundo em Pessoa" que se encontra descrita na Secção 3.1.1. As principais mudanças do sistema irão acontecer na camada *back-end* e *web service* que irão ser descritas nas subsecções seguintes.

4.1.2 Implementação

Back-end

Como referido na Secção 3.1.4, o primeiro protótipo de "O Mundo em Pessoa" tinha um problema de consistência de informação na obra de Fernando Pessoa e por vezes informação duplicada. Para resolver este problema decidiu-se refazer toda a recolha da obra de um único local.

Para esta recolha das obras de Fernando Pessoa é necessária uma pesquisa manual pelas fontes que fornecem a informação desejada sobre essa obra. Para recolher esta informação das fontes são necessários métodos de *web data mining* como os descritos na Secção 2.1.1 .

No caso deste protótipo, a fonte utilizada é o repositório Arquivo Pessoa que contém todos os textos escritos pelo poeta. Este repositório é a atualização de um projeto intitulado MultiPessoa-Labirinto Multimedia[1] coeditado pela Texto Editora e a Casa Fernando Pessoa. Tem como um dos principais objetivos servir de instrumento de investigação ao permitir pesquisas de texto complexas na obra de Fernando Pessoa.

Foi portanto utilizado um programa que utiliza os métodos de *crawling* dentro deste repositório de forma a percorrer todos os *hyperlinks* que correspondem aos textos da obra.

Para cada um dos *hyperlinks* são utilizados os métodos de *wrapping* que vão percorrer o código fonte HTML em busca de determinados campos que contêm a informação necessária sobre as obras: autor, título e conteúdo.

Estes campos têm todos a mesma estrutura: existe um elemento *div* com um campo *id*=autor que contém o nome do autor, um elemento *H1* com o título e por fim um elemento *div* com id=texto-poesia que contém o conteúdo do texto. Para percorrer este código fonte em busca destes elementos HTML, foi utilizada uma ferramenta chamada

[1] Link para o "Arquivo Pessoa": `http://arquivopessoa.net/info`

Xpath[2] que endereça partes de um documento que possua *tags* XML, como é o caso de HTML por Clark et al. [1999].

O Xpath modela um documento XML como uma árvore de nós através de uma expressão regular. Por exemplo, para conseguir recolher o nome do autor era utilizada a seguinte expressão:

$$//div[@id = autor]/text() \tag{4.1}$$

$//div$ significa todas as divs do documento $[@id = autor]$ é um filtro por id, ou seja, $//div[@id = autor]$ significa todos as divs que tiverem o $id = autor$. O $/text()$ é uma noção para devolver o texto que está a seguir à expressão.

Visto que grande parte da obra de Fernando Pessoa é escrita na língua portuguesa é necessário ter atenção aos caracteres especiais da língua portuguesa (acentos, travessões, etc). De forma a não serem perdidos caracteres no armazenamento da obra é necessário preparar tanto a base de dados como a inserção dos dados com a codificação utf-8.

A base de dados, para além da codificação, necessita da adição de todos os campos para o armazenamento dos dados recolhidos da obra. Neste caso irão ser criados campos relativos ao título, conteúdo e *link* na tabela obra e o nome do autor na tabela entidade.

Outro recurso necessário é a lista de termos a serem pesquisados nas redes sociais. Tendo em conta a experiência adquirida com o projeto "O Mundo em Pessoa", verificou-se que grande parte dos utilizadores que faziam citações à obra de Fernando Pessoa colocavam o nome do poeta ou de um dos heterónimos. Alguns utilizadores em vez do nome utilizam *hashtags* para referir o autor. Devido a este fenómeno, para o sistema contabilizar também estas citações, foram adicionados os nomes sem espaços nesta lista.

Assim, a lista de termos será cada um destes nomes, ou seja, "Fernando Pessoa", "fernandopessoa", "Ricardo Reis", "ricardoreis","Alberto Caeiro", "albertocaeiro", "Álvaro de campos", "alvarodecampos", "Bernardo Soares" e "bernardosoares".

Estando a obra e a lista recolhidas e armazenadas na base de dados, o SocialBus vai recolher as mensagens de acordo com a lista e o sistema de deteção de citações vai trabalhar estas mensagens para fazer o mapeamento com a obra.

Como referido na Secção 2.2, foram deixadas algumas configurações em aberto para que este sistema pudesse ser reutilizado de acordo com o contexto. A lista de *stopwords* e *badwords* irá ser a que foi definida como *template* porque os textos da obra estão escritos na língua portuguesa. Sobre as *badwords*, como a obra não contém nenhum termo calunioso optou-se por eliminar as mensagens das redes sociais que contenham estes termos.

O requisito principal solicitado por parte do Sapo Labs para este projeto foi que fosse o mais restrito possível em termos de citações incorretas visto que as mensagens que são consideradas como citação, são posteriormente apresentadas no *front-end*. Para cumprir

[2]Link para XPATH: `http://www.w3schools.com/XPath/`

este requisito é necessário definir um *threshold* para o sistema de *information retrieval* do "Social Impact" que mantenha um valor de *precision* elevado.

Foi então testado este sistema com um conjunto aleatório de 100 mensagens partilhadas do primeiro protótipo de "O Mundo em Pessoa", que utilizadas para a primeira versão deste projeto, de forma a arranjar um valor de *threshold* que obtivesse um valor de *precision* alto. O valor de *threshold* que manteve todas as citações verdadeiras (*true positives*) foi de 1.0. Este valor irá ser avaliado mais à frente na Secção 4.1.3.

Web service

Os *web services* utilizados neste projeto irão ser uma extensão dos *web services* já construídos no "Social Impact" e descritos na Secção 3.3.2.

Para as interrogações feitas pelo *front-end*, irão ser necessárias consultas mais específicas do que as feitas por defeito. Por exemplo, é necessário fazer consultas por autor (saber qual dos heterónimos foi citado), por datas, etc. Também foram adicionados *web services* para fazer alguma estatística sobre o que é citado, por exemplo, quantas citações foram feitas, quais são os textos mais citados, heterónimos, etc.

Todas estas consultas vão resultar numa quantidade cada vez maior de informação ao longo do tempo. Para que a resposta do sistema seja consistente são utilizados *web services* que permitem a navegação dentro desta informação, assim é possível que a resposta contenha apenas um certo número de resultados, definidos pela consulta.

No Anexo A está presente uma tabela que contém todos os *web services* implementados em "O Mundo em Pessoa"assim como a operação HTTP utilizada e uma descrição do que retornam.

4.1.3 Avaliação

Nesta Secção será feita a avaliação do projeto "O Mundo em Pessoa" em várias categorias. Como o sistema central deste projeto e desta tese é o sistema de deteção de citações, que faz a correspondência entre as mensagens partilhadas nas redes sociais e as obras, então a avaliação irá sobretudo recair sobre este sistema.

Para a avaliação do restante projeto irá ser considerada a aceitação pela parte da equipa do Sapo Labs, os testes feitos por esta empresa e a aceitação do público em geral.

Recolha

Com o uso do "SocialBus", para a recolha das mensagens partilhadas nas redes sociais, foram recolhidas 56212 mensagens desde Janeiro de 2014 até Junho de 2014 (6 meses). Todas estas mensagens foram utilizadas pelo sistema de deteção de citações de forma a mapear as obras a estas mensagens. Visto que o valor definido por *threshold* foi 1.0 como explicado na Secção 4.1.2 então foram obtidos os seguintes valores:

Score	Nº Citações	Classificação
≥ 1.0	4720	É citação
≤ 0.5	44325	Não é citação
Entre 1.0 e 0.5	7168	Não é citação
Total	56212	

Tabela 4.1: Tabela dos resultados recolhidos em 6 meses

Precision e recall

Como o sistema que se está a avaliar é um sistema de *information retrieval*, será usada a avaliação habitualmente utilizada para verificar a qualidade dos resultados retornados, as medidas de *precision* e *recall*. Visto a grande quantidade de dados que a base de dados armazena seria impraticável calcular o *precision* e *recall* de todos os dados visto que é necessária uma verificação manual de forma a verificar se o mapeamento feito de cada mensagem com a respetiva obra é a correta. É portanto necessária a recolha de uma amostra da base de dados de forma a fazer esta verificação. Esta amostra contém um total de 200 resultados que estão divididos em quatro categorias:

1. Mensagens do Twitter que foram classificadas como citação.
2. Mensagens do Twitter que não foram classificadas como citação.
3. Mensagens do Facebook que foram classificadas como citação.
4. Mensagens do Facebook que não foram classificadas como citação.

Estas categorias são portanto definidas de acordo com a rede social visto que o tamanho das suas mensagens é diferente (o Twitter apenas permite mensagens com 139 caracteres enquanto que no Facebook não tem qualquer limitação) e de acordo com a classificação que, como explicado na Secção 3.3.2, é considerada citação ou não citação de acordo com o *threshold* definido.

Os resultados com um score muito alto têm uma probabilidade mais alta de estarem corretos do que valores mais baixos. Então, se fossem considerados resultados com score mais alto para as mensagens que são consideradas citação a *precision* calculada iria ser bastante elevada e se fossem considerados resultados com score mais baixo para as mensagens que não são consideradas citação então o *recall* iria ser elevado.

Assim decidiu-se definir outra restrição aos resultados que irão ser avaliados, apenas irão ser considerados os valores fronteira, para as categorias dos classificados como citação irão ser considerados todos os resultados com score entre 2.0 e 1.0 e os que não são considerados citação todos os resultados entre 1.0 e 0.5. Isto faz com que sejam avaliados apenas os resultados críticos, ou seja, que estão perto do valor definido com *threshold*.

Para recolher estes dados é necessária uma consulta à base de dados com as restrições descritas anteriormente. Dos resultados retornados são escolhidos 50 resultados aleatórios

utilizando o método rand() do MySQL. Visto que apenas queremos 200 resultados, para cada categoria iremos ter 50 resultados.

No anexo B, estão presentes tabelas com dados que foram utilizados para esta avaliação, uma ligação para a obra, uma ligação para a citação a essa obra, o score calculado pelo sistema, o tempo que demorou a fazer o mapeamento e a classificação feita manualmente através da observação da obra e a respetiva citação. Esta classificação é feita de acordo com a Tabela 2.1. A partir desta classificação, são usadas as fórmulas descritas nas Secções 2.7 e 2.8.

Através da aplicação das fórmulas à classificação atribuída, obteve-se então 98% de *precision* e 59% de *recall*.

Como referido na Secção 4.1.2, o valor de *threshold* foi definido de forma a que o valor de *precision* deste sistema fosse o mais alto possível, que foi verificado pelo valor perto dos 100%, resultante desta avaliação. Podemos também verificar que o valor de *recall* está algo longe dos 100%. Isto deve-se ao valor definido como *threshold* que tem uma relação direta com a *precision* e o *recall*. Assim quanto maior for o *threshold* maior a *precision* e menor o *recall* e vice-versa.

Foi efetuada outra avaliação com os mesmos dados mas baixando o *threshold* para 0.5 o que faz com que todas mensagens do teste sejam consideradas *true positive* ou *true negative*. Assim como se pode calcular, o *recall* será 100% contudo a *precision* irá cair para os 83%. Visto que pretendemos o maior valor de *precison* possível, o *threshold* mantear-se-á a 1.0.

Tempo de resposta

Outra avaliação feita para este sistema de deteção de citações é o tempo que demora a processar cada mensagem. Para fazer esta avaliação foi contado o tempo que demora o processamento de cada mensagem dos resultados utilizados para a avaliação anterior.

Chegou-se à conclusão, através da avaliação, que a ferramenta faz a correspondência entre a obra e as mensagens das redes sociais num tempo médio de 0.01 segundos (± 0.02).

Divulgação

A avaliação do resto do projeto "O Mundo em Pessoa" foi feita através de testes de usabilidade e segurança efetuados pelo Sapo Labs. Estes testes tiveram como objetivo encontrar possíveis falhas como, por exemplo, eficiência da resposta através dos *web services* e falhas de segurança. Depois de detetados e corrigidos os problemas, foi então lançado o projeto.

O projeto foi exposto no evento Sapo Codebits e no Dia Aberto da FCUL onde se teve a oportunidade de testar o projeto com vários utilizadores e receber algum feedback sobre a utilização do mesmo.

Para além da exposição nos eventos, mereceu também destaque em vários meios da comunicação social não só nacionais como internacionais. As referências para estes destaques estão no anexo C.

4.2 Caso de estudo: "Lusica"

"Lusica" é um projeto de recolha automática de citações de músicas de artistas lusófonos a partir das redes sociais. Este projeto surgiu como continuação dos projetos de colaboração com o Sapo Labs.

O objetivo é produzir um historial da popularidade dos estilos de música típicos da lusofonia (exemplo: fado, samba, etc.) nas redes sociais. Desta forma, o projeto pretende promover a divulgação dos estilos, artistas e músicas pela comunidade.

Para alcançar este objetivo foi desenvolvido um sistema que usa a arquitetura "Social Impact" tal como foi feito em "O Mundo em Pessoa", com a representação dos dados baseada na ideia do projeto "Music Timeline" descrito na Secção 2.3.3. "Lusica" é portanto outra extensão da arquitetura descrita na Secção 3.3.1 com um contexto completamente diferente.

O contexto musical é um bom candidato para a aplicação do "Social Impact" visto que existem várias caraterísticas que poderão ser utilizadas no que consideramos a obra, como, por exemplo, a letra ou o título da música.

4.2.1 Arquitetura

Visto que este projeto é uma nova extensão de contexto da arquitetura "Social Impact" descrita na Secção 3.3.1, esta arquitetura irá ser semelhante.

A Figura 4.2 representa a arquitetura do "Lusica" que, como se pode verificar, contém as mesmas camadas da arquitetura da outra extensão deste sistema, "O Mundo em Pessoa", descrita na Secção 4.1.1. As diferenças entre estas duas arquiteturas estão principalmente no método de recolha de citações das redes sociais.

4.2.2 Implementação

Back-end

Tal como em "O Mundo em Pessoa", em primeiro lugar foi necessário proceder à recolha da obra. A obra, neste contexto, poderá ser ou a letra da música ou o título da mesma.

Para recolher estas informações, em primeiro lugar, foi necessária a definição da amostra necessária visto que a recolha das músicas de todos os artistas musicais lusófonos era inexequível.

O Last Fm[3] é uma base de dados comunitária que contém informações sobre artistas

[3]Link para o Last FM: `http://www.lastfm.pt/`

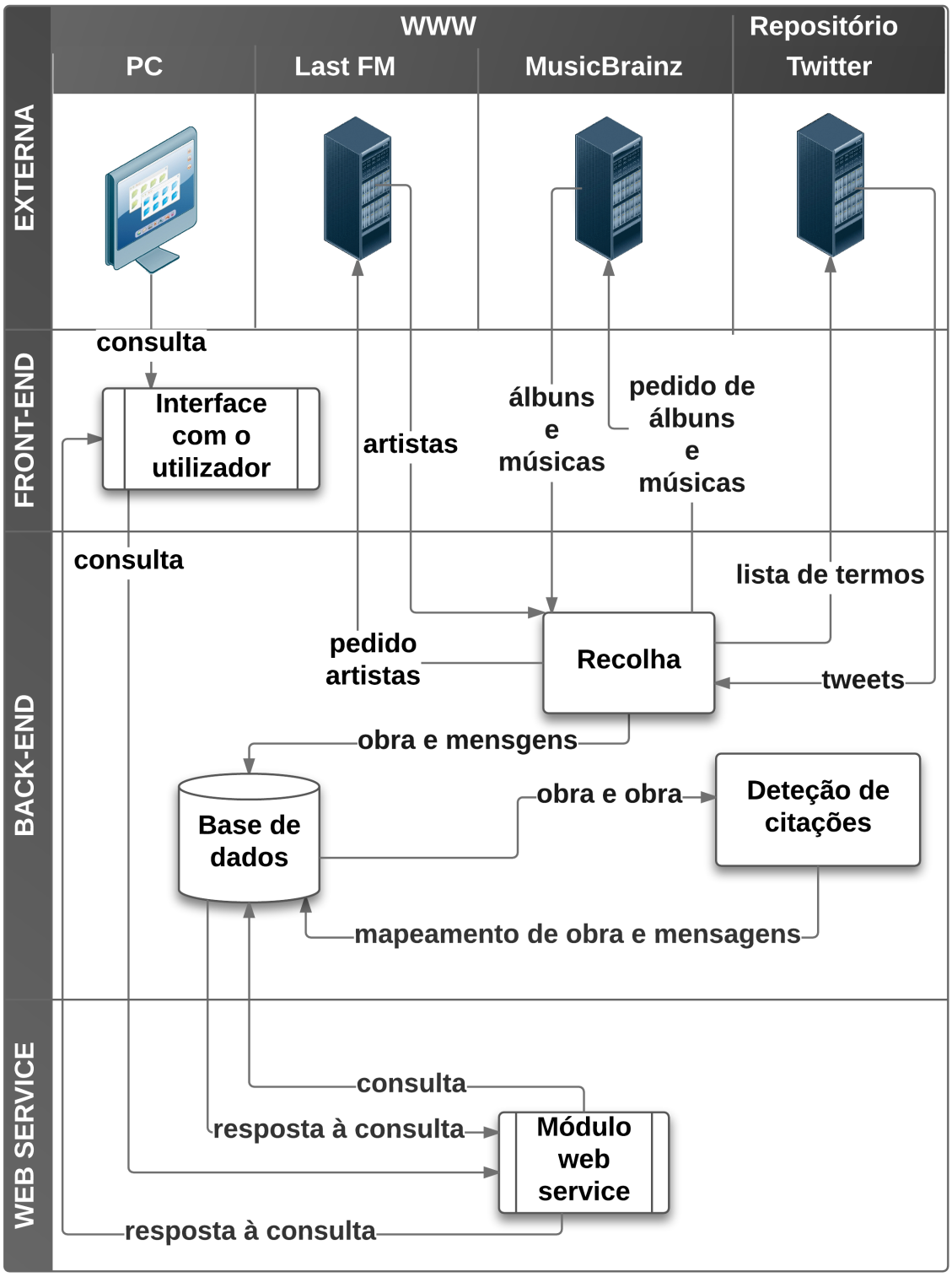

Figura 4.2: Arquitetura do "Lusica"

musicais de todo o mundo. O seu modo de funcionamento é baseado nos princípios da *web* 2.0, como referido na Secção 2.1.4, onde os utilizadores partilham informação sobre

as músicas que ouvem. Assim, é completada a informação que existe na base de dados, o que permite, por exemplo, sugestões ao utilizador sobre músicas tendo em conta os seus gostos musicais.

A principal característica do Last Fm que levou a escolha desta fonte de dados foi o mapeamento dos artistas musicais por *tags* e a existência das *tags* "música lusófona", "música portuguesa", "música brasileira", "música angolana", etc. Estas *tags* devolvem-nos os artistas mais famosos em cada uma destas categorias e por consequente os mais citados nas redes sociais.

Apesar do Last Fm conter bastante informação e com atualização constante, as informações sobre os álbuns e respetivas músicas dos artistas lusófonos nem sempre se encontram completas. Por esta razão foi necessária a consulta a outra fonte de informação, o MusicBrainz.

O MusicBrainz é uma base de dados grátis e *open-source* que tem indexada informação sobre os artistas, álbuns e músicas através dos seus identificadores universais, os mbids. Os mbids são identificadores de 36 caracteres que são permanentemente associados a cada entidade musical: artistas, grupos, álbuns, trabalhos, etc.

Através do MusicBrainz é possível recolher informação de todos os álbuns e respetivas músicas dos artistas recolhidos no Last Fm. Assim recorreu-se à utilização de um sistema de *crawling* e *wrapping* como o descrito na Secção 3.3.2 utilizando a mesma ferramenta, Xpath.

Para a recolha das letras das músicas, foi feita uma pesquisa sobre os vários serviços que forneciam este tipo de informações. Todos estes serviços são baseados em comunidades de utilizadores que partilham as letras das músicas. Isto não garante que estejam corretas ou atualidas visto que não é possivel obter as letras músicas oficiais por razões de direitos de autor. Para além disto, a falta de letras de músicas de artistas lusófonos é também verificável, foram recolhidas letras de várias fontes diferentes e mesmo assim apenas foi conseguido 15% das músicas que estão armazenadas na base de dados. Foi portanto decidido que apenas ia contar o título da música como obra de forma a comparar com as mensagens partilhadas nas redes sociais.

Para recolher informações sobre as redes sociais, era necessário o acesso à informação anterior, para fazer uma retrospetiva ao longo do tempo do que era comentado nas redes sociais sobre a música lusófona. Por isso, não basta a utilização do "SocialBus" como em "O Mundo em Pessoa", visto que é necessária a informação das mensagens partilhadas ao longo do tempo.

De forma a recolher as informações anteriores foi utilizado um repositório de *tweets* disponibilizados pela mesma equipa que criou o SocialBus. Este repositório tem todas as mensagens partilhadas no Twitter de uma comunidade de utilizadores portugueses desde 2011.

Visto que não foi encontrado nenhum repositório semelhante para o Facebook, foi de-

cidido substituir esta rede social por uma dedicada à música e que não tivesse limitações no acesso aos comentários. Por isso, foram recolhidos os comentários dos artistas lusófonos do Last Fm. A base de dados irá ser alterada para este contexto visto que será preciso guardar mais informações sobre as músicas: álbuns, artistas e músicas.

O sistema de deteção de citações funciona de maneira diferente de "O Mundo em Pessoa" por causa de especificidade do contexto. Como referido anteriormente, os títulos das músicas serão considerados como obras. Se fosse utilizado o mesmo método do que o descrito na Secção 2.2, então eram comparados todos os títulos com uma mensagem das redes sociais. Como existem músicas diferentes com títulos iguais, mas de autores diferentes, foi verificado que se forem utilizados todos os títulos como obras irá resultar num valor de *precision* baixa.

Foi então necessária uma modificação neste sistema de deteção de citações de forma a resolver este problema. Assim para cada mensagem apenas serão carregados os títulos das músicas correspondentes ao artista citado de forma a garantir que o *searcher* só mapeie as obras relativas ao artista citado. Sempre que uma nova mensagem entrar no sistema, os documentos serão atualizados, apagando os anteriores e adicionando os que correspondem ao artista citado.

Com o sistema concluído, à semelhança do que aconteceu em "O Mundo em Pessoa", a empresa Sapo Labs solicitou que os dados fossem o mais precisos possível. Então, foi feita uma análise aos resultados através do cálculo dos valores de *precision*. Esta análise resultou num valor de *precision* por volta dos 70% o que é um valor baixo visto que os valores de *precision* de "O Mundo em Pessoa" chegavam aos 98%. Verificou-se que o baixo valor de *precision* devia-se ao facto de haver músicas cujo título era igual ao nome do artista correspondente. Por exemplo, existe uma música da banda Da Weasel que tem uma música chamada Da Weasel e se uma mensagem partilhada nas redes sociais contivesse apenas Da Weasel era considerada como citada, embora o utilizador não se estivesse a referir à música.

Para resolver este problema decidiu-se não contar estes casos quando o utilizador apenas se refere ao nome do artista. Os resultados desta modificação serão analisados mais à frente na seccção 4.2.3.

Web services

À semelhança do que aconteceu em "O Mundo em Pessoa", os *web services* utilizados neste projeto irão ser uma extensão dos *web services* já construídos no "Social Impact" e descritos na Secção 3.3.2. Para as interrogações feitas pelo *front-end*, irão ser necessárias consultas mais específicas para este contexto. Neste caso iremos ter *web services* para devolver os álbuns musicais citados de uma certa fonte, data e estilo, e *web services* para devolver tops e pesquisar nos mesmos. O Anexo D contém uma tabela com todos os *web services* disponíveis para o "Lusica".

Front-end

Através da informação fornecida pelos *web services*, foi construída uma interface do utilizador tal como em "O Mundo em Pessoa".

A apresentação da informação nesta interface foi baseada na representação utilizada no projeto "Music Timeline" descrito na Secção 2.3.3, que irá conter uma *timeline* que mostra, ao longo do tempo, qual o estilo de música mais falado nas redes sociais.

Em primeiro lugar foi construído um gráfico com os dados recolhidos e respetivamente mapeados, com a música correspondente. Através da música conseguimos facilmente descobrir o estilo e assim criar um gráfico que mostra estes dados.

Para a construção deste gráfico contou-se com a ajuda da biblioteca Stacked Graphs do D3[4] que permite associar dados a um Document Object Model(DOM) e assim aplicar transformações a um documento como, por exemplo, a geração de gráficos numa página HTML como explica o artigo de Bostock et al. [2011].

Na Figura 4.3 está representada uma página do "Lusica" onde se pode verificar o gráfico, no qual cada camada representa um estilo musical. Em cima do gráfico estão os vários estilos musicais que serão destacados cada vez que o cursor passar em cima da camada correspondente. Caso o utilizador faça *click* em cima de uma camada e num determinado mês, serão apresentados os álbuns desse estilo e que foram citados nesse mês ordenados descendentemente por número de citações e o estilo musical correspondente ficará selecionado.

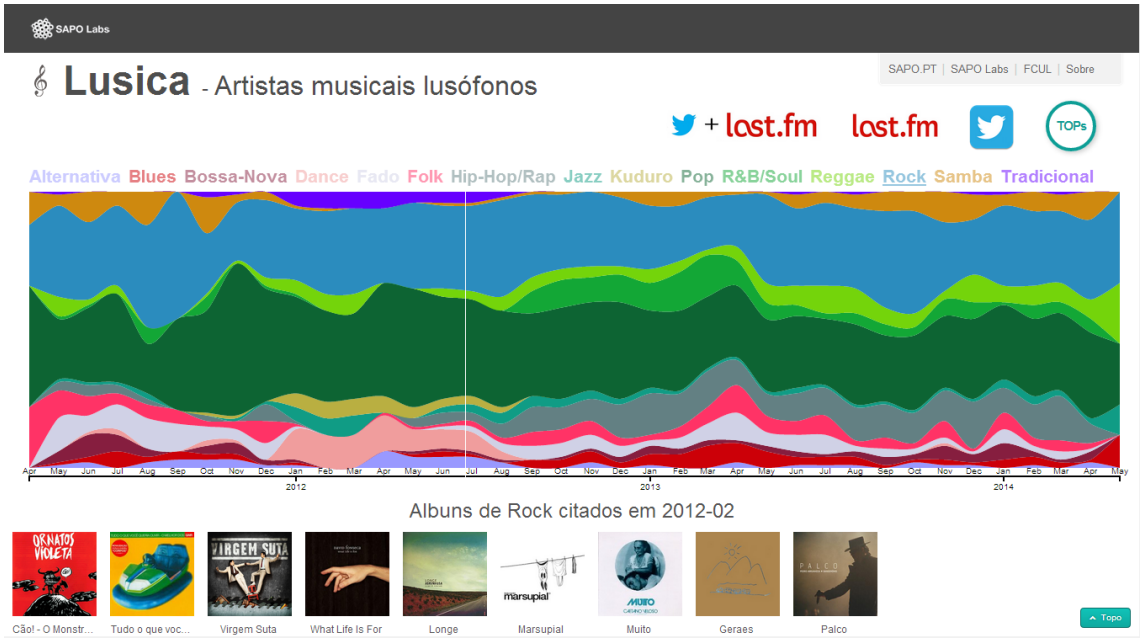

Figura 4.3: Página principal do "Lusica"

[4]Link para o D3: `http://d3js.org/`

Para os dados relativos aos comentários no Last Fm é feito exatamente o mesmo procedimento. É também disponibilizado um gráfico relativo às mensagens partilhadas nestas duas redes sociais em conjunto. É ainda disponibilizada uma funcionalidade que permite ao utilizador efetuar consultas cujos dados serão retornados em forma de top, como se pode verificar na Figura 4.4. Isto permitirá aos utilizadores consultar quais os artistas, álbuns e músicas mais citados numa determinada data definida. Este top é comparado com os dados de noutra data também definida pelo utilizador, posteriormente é feita a comparação de forma a dizer se cada artista, álbum ou música desceram de posição, subiram, mantiveram-se ou é uma nova entrada no top.

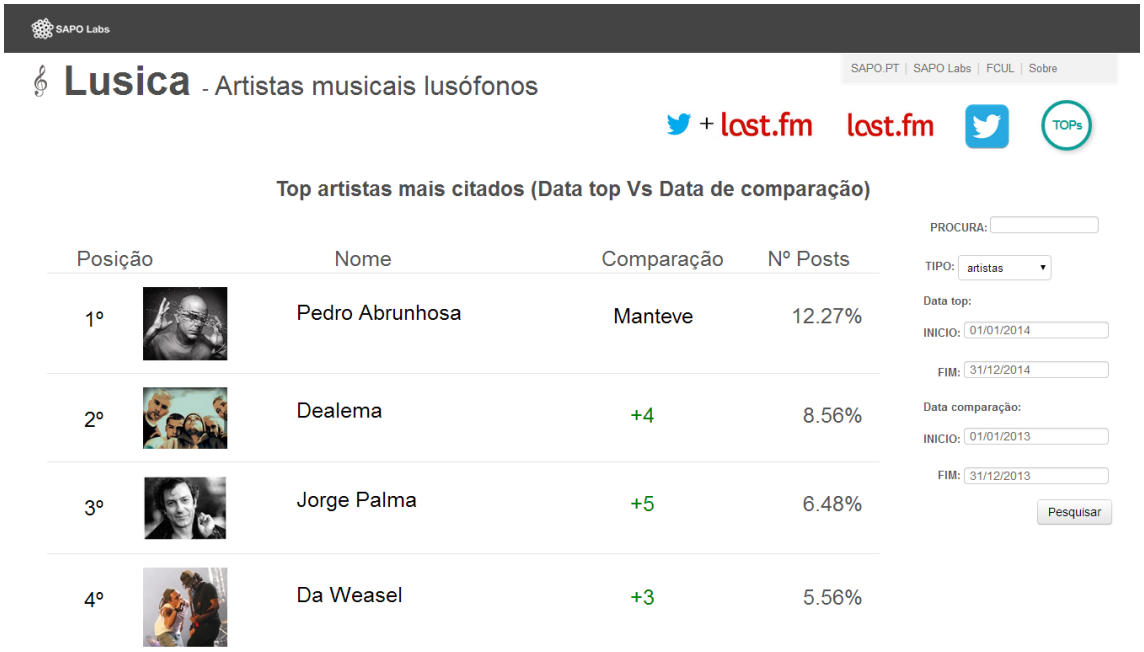

Figura 4.4: Tops do "Lusica"

4.2.3 Avaliação

À semelhança do que foi feito para a avaliação do projeto "O Mundo em Pessoa" irão ser aplicadas medidas de avaliação ao sistema de deteção de citações aplicado ao contexto do "Lusica" e a aceitação pelo Sapo Labs e o público em geral.

Recolha

Para o processo do sistema de deteção de citações foram consideradas 423612 tweets partilhados desde Janeiro de 2001 até Junho de 2014. O valor definido de *threshold* foi de 1.0 utilizando "O Mundo em Pessoa" como referência e os valores obtidos foram os apresentados na Tabela 4.2.

Score	Nº Citações	Classificação
≥ 1.0	7628	É citação
≤ 0.5	408106	Não é citação
Entre 1.0 e 0.5	7878	Não é citação
Total	423612	

Tabela 4.2: Tabela dos resultados do "Lusica"

Precision e recall

Nesta avaliação irão ser utilizados os mesmos métodos de avaliação utilizados para o "O Mundo em Pessoa" descritos na Secção 4.1.3.

Foi portanto recolhida uma amostra de 100 resultados em que 50 são mensagens classificadas como citação e 50 classificadas como não citação. À semelhança do que aconteceu com "O Mundo em Pessoa" é necessária uma consulta à base de dados com as restrições necessárias. Dos resultados retornados são escolhidos 50 resultados aleatórios utilizando o método rand() do MySQL.

No anexo E, estão presentes tabelas com dados que foram utilizados para esta avaliação, o nome do artista e da música citada, uma ligação para a citação a essa música, o *score* calculado pelo sistema, o tempo que demorou a fazer o mapeamento e a classificação feita manualmente através da observação da obra e a respetiva citação. Esta classificação é feita de acordo com a Tabela 2.1. De acordo com a classificação, são usadas as fórmulas 2.7 e 2.8.

Através da aplicação das fórmulas à classificação atribuída, obteve-se então 100% de *precision* e 53% de *recall*.

O valor *threshold* foi definido com base no valor definido para o projeto "O Mundo em Pessoa". Com isto obteve-se uma *precision* de 100% para esta amostra e um *recall* um pouco mais baixo que no projeto anterior. Isto deve-se sobretudo ao tamanho do que é considerado obra (neste caso, os títulos das músicas). Enquanto que em "O Mundo em Pessoa" praticamente todos os textos eram extensos, os títulos das músicas a maior parte das vezes não o são (por vezes é apenas uma palavra).

Tempo de resposta

Outra avaliação feita para este sistema de deteção de citações é o tempo que demora a processar cada mensagem. Para fazer esta avaliação foi contado o tempo que demora o processamento de cada mensagem dos resultados utilizados para a avaliação anterior.

Chegou-se à conclusão, através da avaliação, que a ferramenta faz a correspondência entre a obra e as mensagens das redes sociais num tempo médio de 0.02 segundos. Comparando com os valores de "O Mundo em Pessoa" (0.01), este acréscimo do tempo deve-se à alteração efetuada no sistema de *information retrieval* de forma a carregar as músicas

do autor que está a ser citado, ou seja, por cada mensagem partilhada recolhida, será necessário atualizar a obra.

Divulgação

À semelhança do que aconteceu em "O Mundo em Pessoa", a avaliação do resto do projeto foi feita através de testes de usabilidade e segurança efetuados pelo Sapo Labs.

O projeto foi também exposto no evento Sapo Codebits e no Dia Aberto da FCUL onde se teve a oportunidade de testar o projeto com vários utilizadores e receber algum feedback sobre a utilização do mesmo.

No evento Sapo Codebits, o "Lusica" teve destaque nos meios de comunicação social mais concretamente nos canais de televisão RTP e SIC. Os *links* para esta divulgação do projeto encontram-se no anexo E.

4.3 "Onde há bola"

O "Onde há bola" foi um projeto desenvolvido por alunos da cadeira de Aplicações na Web do Mestrado em engenharia informática da Faculdade de Ciências da Universidade de Lisboa e também foi um projeto com a colaboração do Sapo Labs.

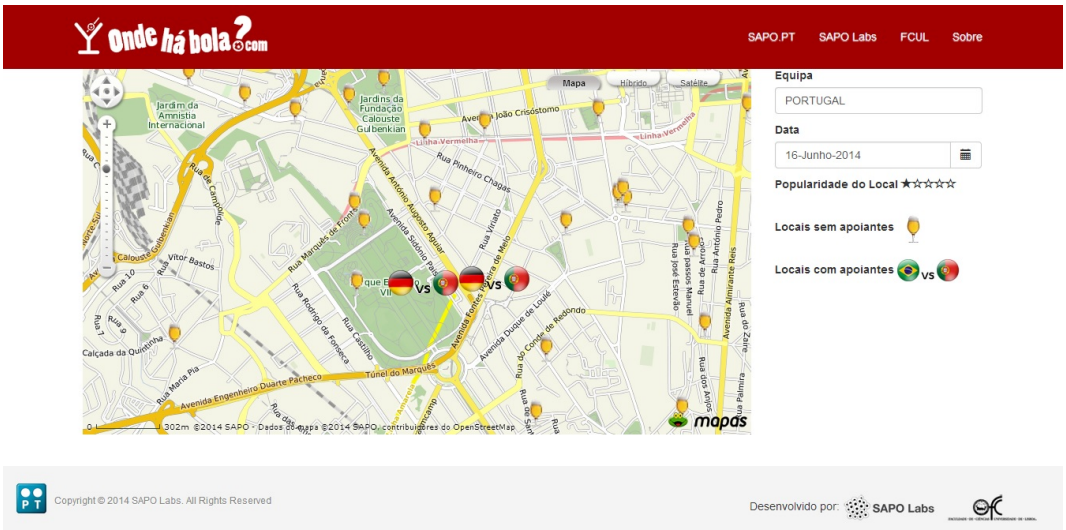

Figura 4.5: Página principal de "Onde há bola"

O principal objetivo do "Onde há bola" foi indicar os locais onde se pode assistir aos jogos do Mundial de Futebol - Brasil 2014, e o número de pessoas que estão a apoiar cada uma das equipas em cada um desses locais. O utilizador pode assim escolher um local perto de si para apoiar a sua equipa favorita num ambiente mais favorável, ao mesmo tempo que pode indicar à comunidade a sua intenção de assistir ao jogo e a equipa que irá apoiar.

Este projeto não é um caso de uso da arquitetura "Social Impact" porque apenas alguns elementos desta arquitetura foram reutilizados, mais especificamente, estrutura da base de dados e dos *web services* mas, foi sobretudo utilizado o conhecimento adquirido da experiência com os projetos anteriormente implementados.

A Figura 4.5 mostra o resultado da pesquisa da equipa portuguesa no dia 16 de Junho. Neste dia aconteceu o jogo Alemanha-Portugal que é mostrado no mapa nos sítios onde foi transmitido este jogo.

4.4 "Missinks"

O "Missinks", à semelhança do "Onde há bola", é um projeto que não é um caso de uso da arquitetura "Social Impact" mas utiliza a estrutura da base de dados e dos *web services*.

Este projeto é uma aplicação web que dada uma consulta identifica os *links* que estão nas primeira duas páginas de resultados do Google de um determinado país à escolha comparando com as quatro primeiras páginas de resultados do Google de outro país e devolvendo os *links* diferentes. A motivação por detrás deste projeto foi criar uma ferramenta que ajudasse a encontrar os links removidos segundo a legislação de proteção de dados do Google[5] que permite aos utilizadores removerem links de acordo com a pesquisa pelo seu nome.

Figura 4.6: Consulta no "Missinks"

[5]Link para a informação sobre a lesgislação: `https://support.google.com/websearch/answer/2744324`

A Figura 4.6 mostra a interface do "Missinks". Esta é uma interface bastante simples que apenas contém uma caixa de texto que permite fazer a consulta e duas *select boxes* que permitem selecionar os domínios Google. Os *links* resultantes serão apresentados abaixo das *select boxes*. O exemplo apresentado pela figura mostra uma pesquisa por "Carlos Barata", ou seja, será procurado "Carlos Barata" no Google canadiano (.ca) e Google português (.pt), serão recolhidas as primeiras duas páginas de resultados do canadiano e as quatro primeiras do português. Os *links* das páginas de resultados retornados pelo Google canadiano serão comparados com os links do português e os que não se encontrarem nos links portugueses serão retornados e mostrados em baixo das *select boxes*.

Capítulo 5

Conclusão

Nesta tese foi apresentada uma abordagem para fazer a correspondência entre mensagens partilhadas nas redes sociais que citam uma determinada obra, e as respetivas obras originais. Esta abordagem passou pela construção de um primeiro protótipo, "O Mundo em Pessoa" que apenas estava focado no contexto da obra de Fernando Pessoa. Este protótipo foi feito de forma a cumprir o prazo do aniversário do poeta. Para além dos vários problemas detetados com este protótipo, havia a necessidade da abstração da abordagem de forma a ser aplicado a vários contextos. Foi portanto criada a arquitetura "Social Impact" que, dada uma obra e uma lista de termos de procura, faz a recolha das mensagens partilhadas nas redes sociais que refiram algum dos termos da lista de termos de procura, recorrendo a ferramenta "SocialBus". Depois de recolhidas as mensagens, estas são mapeadas com a obra citada, recorrendo à ferramenta Apache Lucene.

Para obter uma versão melhorada de "O Mundo em Pessoa", foi criado o segundo protótipo deste projeto que implementa a arquitetura "Social Impact" e sendo assim é um caso de uso desta arquitetura. Como o objetivo da construção do "Social Impact" passava também pela abstração a vários contextos, foi criado o "Lusica" que é também um caso de uso do "Social Impact", neste caso no contexto da música lusófona. Para além destes casos de uso foram criados mais dois projetos "Onde há bola" e "Missinks" que não são casos de uso do "Social Impact" mas usam alguns componentes desta arquitetura.

Ambos os casos de uso foram submetidos a uma avaliação onde foi concluído que grande parte das mensagens recolhidas não são consideradas citação, após o mapeamento das citações com a obra, o que tem uma relação direta com os valores de *precision*, que são perto de 100% (valor ideal), e de *recall* que é baixo. Foi também efetuada uma avaliação do tempo de resposta médio do mapeamento das mensagens partilhadas nas redes sociais com as obra citadas, o que se revelou não ser relevante tendo em conta a perspetiva do utilizador, ou seja, o tempo médio que leva a fazer este mapeamento é menor que um segundo o que não provoca um grande atraso entre o tempo que a citação é recolhida e é mostrada ao utilizador.

O que define as mensagens serem citação ou não é o *threshold* que é diretamente

proporcional ao valor de *precision* e inversamente proporcional ao valor de *recall*. Assim, o valor definido de *threshold* foi o que manteve a maior *precision* possível mantendo um número aceitável de citações.

Os projetos "O Mundo em Pessoa"e "Lusica" foram testados por equipas da empresa Sapo Labs que sempre deu o apoio necessário para a concretização destes projetos. Após os testes e respetiva correção de erros técnicos procedeu-se à divulgação e apresentação dos projetos em vários eventos. Devido a estas apresentações, o projeto mereceu destaque nos meios de comunicação favorecendo não só o bom nome da equipa Sapo Labs da PT como também o da Faculdade de Ciências da Universidade de Lisboa.

Como trabalho futuro sugere-se a implementação de um sistema automático para definir o *threshold*. A secção 2.2 mostra que num sistema como o implementado é possível ter uma função que permita o feedback do utilizador sobre os resultados retornados. Tal possibilitaria a criação de um classificador que permitisse definir automaticamente o *threshold* de acordo com o feedback dos utilizadores.

Para o resto do sistema, seria interessante aproveitar o sistema de feedback de forma a permitir ao utilizador inserir informação de acordo com os padrões da web 2.0. Por exemplo, o "Lusica" poderia tornar-se uma espécie de rede social onde os utilizadores poderiam não só sugerir músicas, artistas, letras, etc, como também definir preferências e sugerir outras músicas e artistas de acordo com os gostos pessoais.

A nível pessoal esta tese permitiu-me o contacto direto com uma empresa, o que foi uma experiência enriquecedora sobretudo a nível de processos empresariais como por exemplo, o trabalho sobre prazos curtos, tomadas de decisão e metodologias empresariais. Para além desta experiência foi também interessante fazer a ponte do meio académico com o meio empresarial visto que, em todos os projetos realizados, estiveram envolvidas equipas de ambos os meios.

Com o envolvimento nos projetos "Onde há bola" e "Missinks" ganhei outro tipo de experiência visto que para além de estar envolvido em tarefas de implementação, tive também responsabilidade nas tarefas de coordenação.

Em termos do desenvolvimento dos projetos foram cumpridos os objetivos definidos inicialmente. Esta tese foi um bom exemplo de como a ligação entre as empresas e mundo académico pode resultar na criação, transmissão e difusão da cultura com base na ciência e tecnologia.

Apêndice A

Tabela de *web services* de "O Mundo em Pessoa"

Operação	Consulta	Retorno
GET	obra/	retorna a informação sobre todas as obras que se encontram na base de dados
GET	obra/start/{number}/limit /{number}/	retorna a informação sobre todas as obras permitindo a navegação entre esta informação.
GET	obra/{id}/	retorna a informação sobre a obra que corresponde ao id passado na consulta
GET	obra/{id}/citacao/	retorna a informação sobre as mensagens partilhadas nas redes sociais que foram consideradas citações, pelo sistema, à obra que corresponde ao id
GET	obra/{id}/citacao/start/ {number}/limit/{number}/	retorna a informação sobre as mensagens partilhadas nas redes sociais que foram consideradas citações, pelo sistema, à obra que corresponde ao id permitindo navegação.
GET	obra/autor/{id}/	retorna a informação sobre as obras do autor que corresponde ao id.

GET	obra/autor/{id}/start/{number}/limit/{number}/	retorna a informação sobre as obras do autor que corresponde ao id. O start e limit permitem fazer a navegação entre a informação disponibilizada.
GET	citacao/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema.
GET	citacao/{id}/	retorna a informação sobre a mensagem correspondente ao id partilhada nas redes sociais que foi considerada citação pelo sistema.
GET	citacao/{id}/texto/	retorna a informação sobre a o texto que está mapeado à mensagem correspondente ao id partilhada nas redes sociais que foi considerada citação pelo sistema.
GET	citacao/from/{data}/to/{data}/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema que estejam dentro do intervalo de datas passado.
GET	citacao/start/{number}/limit/{number}/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas permitindo a navegação entre esta informação.
GET	citacao/from/{data}/to/{data}/start/{number}/limit/{number}/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema que estejam dentro do intervalo de datas passado permitindo a navegação entre esta informação.

GET	citacao/autor/{id}/from/{data}/to/{data}/start/{number}/limit/{number}/	retorna a informação sobre todas as mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema, que estejam dentro do intervalo de datas passado e cujo id correspondente ao autor do texto mapeado com a citação seja o passado, permitindo a navegação entre esta informação.
GET	citacao/from/{data}/to/{data}/texto/{id}/	retorna a informação sobre todos os textos mapeados às mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema que estejam dentro do intervalo de datas passado.
GET	citacao/from/{data}/to/{data}/texto/{id}/start/{number}/limit/{number}/	retorna a informação sobre todos os textos mapeados às mensagens partilhadas nas redes sociais que foram consideradas citação pelo sistema que estejam dentro do intervalo de datas passado, permitindo a navegação entre esta informação.
GET	utilizador/{id}/	retorna a informação sobre o utilizador com o id passado.
GET	palavrascitadas/	retorna a informação sobre o mapemento de todos as mensagens das redes sociais com a obra respetiva.
GET	palavrascitadas/{id}	retorna a informação sobre o mapemento da mensagem, com o id correspondente, das redes sociais com a obra respetiva.
GET	palavrascitadas/start/{number}/limit/{number}/	retorna a informação sobre o mapemento de todos as mensagens das redes sociais com a obra respetiva, permitindo a navegação entre esta informação.

GET	estatistica/texto/from/ {data}/to/{data}/	retorna o numero de textos citados entre o intervalo de datas passado.
GET	estatistica/texto/{id}/ from/{data}/to/{data}/	retorna o numero de vezes que o texto com o id correspondente ao passado é citado entre o intervalo de datas passado.
GET	estatistica/texto/start/ {number}/limit/{number}/	retorna o numero de textos citados, permitindo a navegação entre esta informação.
GET	estatistica/texto/from/ {data}/to/{data}/start/ {number}/limit/{number}/	retorna o numero de textos citados entre o intervalo de datas passado, permitindo a navegação entre esta informação.
GET	estatistica/autor/	retorna o numero de textos citados ordenados por autor.
GET	estatistica/autor/{id}/	retorna o numero de textos citados do autor cujo id corresponde ao id passado.
GET	estatistica/autor/from/ {data}/to/{data}/	retorna o numero de textos citados ordenados por autor dentro do intervalo de datas passado.
GET	estatistica/autor/{id}/ from/{data}/to/{data}/	retorna o numero de textos citados de um autor dentro do intervalo de datas passado.
GET	estatistica/autor/start/ {number}/limit/{number}/	retorna o numero de textos citados ordenados por autor, permitindo a navegação entre esta informação.
GET	estatistica/autor/from/ {data}/to/{data}/start/ {number}/limit/{number}/	retorna o numero de textos citados ordenados por autor dentro do intervalo de datas passado, permitindo a navegação entre esta informação.

Apêndice B

Tabelas de avaliação de "O Mundo em Pessoa"

B.1 Mensagens do Facebook que foram classificadas como citação

Obra	Citação	Score	Tempo	Class
arquivopessoa.net/textos/2206	facebook.com/100001757013200_545059055562657	1.07927	0.086	TP
arquivopessoa.net/textos/344	facebook.com/100002323087670_538397202914378	1.21108	0.041	TP
arquivopessoa.net/textos/4468	facebook.com/1028843850_10200967154074268	1.52256	0.017	TP
arquivopessoa.net/textos/4457	facebook.com/1790805876_10200556786520734	1.45993	0.009	TP
arquivopessoa.net/textos/3339	facebook.com/100002670659235_547312392034444	1.78546	0.024	TP
arquivopessoa.net/textos/263	facebook.com/100004058120879_420212014790732	1.0927	0.002	TP
arquivopessoa.net/textos/1463	facebook.com/100002980941169_515895635186463	1.23603	0.023	TP
arquivopessoa.net/textos/1122	facebook.com/100000870902198_679169422122082	1.3306	0.01	TP

arquivopessoa.net/ textos/1463	facebook.com/ 1423175558_ 10203557647808137	1.17214	0.025	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100001716806598_ 644277432306144	1.19664	0.011	TP
arquivopessoa.net/ textos/2695	facebook.com/ 100002328586491_ 615504458537168	1.27865	0.02	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100002118027047_ 626891874058074	1.74958	0.047	TP
arquivopessoa.net/ textos/1497	facebook.com/ 100002091115052_ 603701759709522	1.07843	0.086	FP
arquivopessoa.net/ textos/1463	facebook.com/ 100002438541221_ 606191562805456	1.10445	0.031	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100003508165944_ 485052058288395	1.2865	0.043	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100002671176063_ 563201263778908	1.57279	0.042	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100003069123826_ 499668250145475	1.39507	0.041	TP
arquivopessoa.net/ textos/1275	facebook.com/ 100003781295337_ 411085962360777	1.30753	0.007	TP
arquivopessoa.net/ textos/1817	facebook.com/ 100001758591247_ 598815100187104	1.62541	0.026	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100000713891641_ 739440079423115	1.64563	0.014	TP
arquivopessoa.net/ textos/2492	facebook.com/ 1594124949_ 10201388877155924	1.06539	0.074	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100002600209341_ 578629372233731	1.62791	0.014	TP
arquivopessoa.net/ textos/4369	facebook.com/ 100005770689805_ 227327037469651	1.8238	0.005	TP

arquivopessoa.net/ textos/1463	facebook.com/ 100001942651168_ 615705288504221	1.16599	0.009	TP
arquivopessoa.net/ textos/1709	facebook.com/ 100003681731617_ 501882546611132	1.00044	0.01	TP
arquivopessoa.net/ textos/1463	facebook.com/ 1211467567_ 10201582555104997	1.45991	0.02	TP
arquivopessoa.net/ textos/163	facebook.com/ 100002855662518_ 498846063553911	1.05467	0.104	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100000005922140_ 770549539621903	1.16599	0.009	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100003613387332_ 449847531812356	1.12744	0.005	TP
arquivopessoa.net/ textos/163	facebook.com/ 100004002370671_ 425176577625751	1.04458	0.064	TP
arquivopessoa.net/ textos/4056	facebook.com/ 100005432407295_ 236284409895971	1.84584	0.007	TP
arquivopessoa.net/ textos/3640	facebook.com/ 100005139910619_ 212216115626384	1.09465	0.014	TP
arquivopessoa.net/ textos/1463	facebook.com/ 1399808150_ 10203487102163761	1.4407	0.012	TP
arquivopessoa.net/ textos/4468	facebook.com/ 100000051464668_ 760950020583373	1.15842	0.01	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100006314882564_ 1500356440184838	1.27535	0.016	TP
arquivopessoa.net/ textos/632	facebook.com/ 654403485_ 10152313821368486	1.98343	0.021	TP
arquivopessoa.net/ textos/19	facebook.com/ 100003510177632_ 474242502702751	1.08719	0.052	TP
arquivopessoa.net/ textos/1463	facebook.com/ 285111658321686_ 285154101650775	1.47355	0.012	TP

arquivopessoa.net/ textos/1463	facebook.com/ 100000965130133_ 701240993251399	1.56776	0.018	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100001396905118_ 688526294537261	1.0267	0.012	TP
arquivopessoa.net/ textos/2375	facebook.com/ 186345981388240_ 749967008359465	1.2457	0.007	TP
arquivopessoa.net/ textos/1122	facebook.com/ 100004793804057_ 282693011900490	1.76137	0.006	TP
arquivopessoa.net/ textos/4522	facebook.com/ 1676954623_ 10201792137639945	1.09629	0.03	TP
arquivopessoa.net/ textos/2567	facebook.com/ 1637355949_ 10203182968128518	1.15125	0.014	TP
arquivopessoa.net/ textos/1655	facebook.com/ 863016237058243_ 864187940274406	1.00485	0.013	TP
arquivopessoa.net/ textos/1122	facebook.com/ 100000432498045_ 815467531810989	1.9728	0.007	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100001569691631_ 746179065444410	1.80488	0.012	TP
arquivopessoa.net/ textos/1522	facebook.com/ 191231477743626_ 276805589148496	1.00288	0.013	TP
arquivopessoa.net/ textos/4056	facebook.com/ 100003991118545_ 422344234575277	1.06734	0.003	TP
arquivopessoa.net/ textos/1463	facebook.com/ 100003361102800_ 546555458799819	1.89882	0.002	TP

B.2 Mensagens do Facebook que não foram classificadas como citação

Obra	Mensagem	Score	tempo	Class
arquivopessoa.net/ textos/1463	facebook.com/ 100006248011502_ 1421694201382167	0.678247	0.063	FN

arquivopessoa.net/ textos/2515	facebook.com/ 100000419082200_ 764306426926653	0.565943	0.016	TN
arquivopessoa.net/ textos/1463	facebook.com/ 100000506328618_ 747306308629573	0.881248	0.009	FN
arquivopessoa.net/ textos/2375	facebook.com/ 100001692440574_ 647622968637445	0.535488	0.008	FN
arquivopessoa.net/ textos/2131	facebook.com/ 100001682124876_ 655760647823333	0.531672	0.012	FN
arquivopessoa.net/ textos/1275	facebook.com/ 100000762412068_ 635654336469974	0.547471	0.06	FN
arquivopessoa.net/ textos/655	facebook.com/ 100007758070835_ 1396499467285309	0.508393	0.006	TN
arquivopessoa.net/ textos/1463	facebook.com/ 100006427419384_ 1556166827940883	0.625522	0.004	FN
arquivopessoa.net/ textos/84	facebook.com/ 100002654877483_ 548527015245789	0.869132	0.055	FN
arquivopessoa.net/ textos/4323	facebook.com/ 100000839471197_ 652660551438588	0.831054	0.046	FN
arquivopessoa.net/ textos/2978	facebook.com/ 554320251_ 10153913745555252	0.764987	0.014	FN
arquivopessoa.net/ textos/163	facebook.com/ 100001336250833_ 653014914753044	0.530907	0.013	FN
arquivopessoa.net/ textos/348	facebook.com/ 100003319730851_ 511730088947644	0.618759	0.021	FN
arquivopessoa.net/ textos/84	facebook.com/ 100002737711175_ 463564583744795	0.950287	0.023	FN
arquivopessoa.net/ textos/1463	facebook.com/ 100007929604588_ 1388869708054003	0.990088	0.004	FN
arquivopessoa.net/ textos/821	facebook.com/ 100001097806664_ 681680601878527	0.730237	0.01	FN

arquivopessoa.net/ textos/2978	facebook.com/ 100000434513961_ 781269311897545	0.764987	0.011	FN
arquivopessoa.net/ textos/4409	facebook.com/ 100001707965272_ 657312577669002	0.790822	0.007	FN
arquivopessoa.net/ textos/1948	facebook.com/ 100003779864220_ 421955374607121	0.565847	0.003	TN
arquivopessoa.net/ textos/821	facebook.com/ 100001525972788_ 688905897836936	0.730237	0.014	FN
arquivopessoa.net/ textos/4323	facebook.com/ 100002261499165_ 621967161221995	0.681074	0.036	FN
arquivopessoa.net/ textos/1759	facebook.com/ 100002036461491_ 614992231911987	0.575187	0.002	TN
arquivopessoa.net/ textos/4264	facebook.com/ 100002724747812_ 476469442453876	0.506856	0.003	TN
arquivopessoa.net/ textos/1463	facebook.com/ 100002559320906_ 581244351970870	0.542667	0.006	FN
arquivopessoa.net/ textos/888	facebook.com/ 100003244009037_ 582322005219261	0.545749	0.014	TN
arquivopessoa.net/ textos/827	facebook.com/ 1525370766_ 10201672418321984	0.829285	0.009	FN
arquivopessoa.net/ textos/4468	facebook.com/ 100002065016346_ 605891406156354	0.666121	0.006	FN
arquivopessoa.net/ textos/2811	facebook.com/ 147282901999088_ 680424912018215	0.527974	0.006	TN
arquivopessoa.net/ textos/2556	facebook.com/ 100003229359017_ 532986090152365	0.838101	0.002	FN
arquivopessoa.net/ textos/3749	facebook.com/ 100006390581388_ 1486513511571654	0.53583	0.001	TN
arquivopessoa.net/ textos/4323	facebook.com/ 100002363469284_ 630384683716975	0.969551	0.021	FN

arquivopessoa.net/ textos/1463	facebook.com/ 1139179143_ 10201664035299541	0.79918	0.005	FN
arquivopessoa.net/ textos/2269	facebook.com/ 100006005737158_ 211062849103905	0.816215	0.006	FN
arquivopessoa.net/ textos/3364	facebook.com/ 100000777619662_ 653837384652176	0.521936	0.007	FN
arquivopessoa.net/ textos/1463	facebook.com/ 100000204965585_ 856012004415624	0.67745	0.006	FN
arquivopessoa.net/ textos/3458	facebook.com/ 100005471436000_ 235950259930687	0.760826	0.006	FN
arquivopessoa.net/ textos/1985	facebook.com/ 100002579710784_ 583970881698897	0.506426	0.004	TN
arquivopessoa.net/ textos/4468	facebook.com/ 100007131521080_ 1432487296998974	0.840732	0.012	FN
arquivopessoa.net/ textos/1463	facebook.com/ 100004969711121_ 282164801959175	0.67745	0.003	FN
arquivopessoa.net/ textos/2222	facebook.com/ 100004471350910_ 320888411403535	0.665014	0.01	TN
arquivopessoa.net/ textos/1463	facebook.com/ 100000955567038_ 728730107168835	0.679699	0.003	FN
arquivopessoa.net/ textos/2562	facebook.com/ 100002383093845_ 655480484541382	0.577371	0.002	TN
arquivopessoa.net/ textos/2567	facebook.com/ 100007423231758_ 1465845413672886	0.611499	0.002	FN
arquivopessoa.net/ textos/1463	facebook.com/ 100000537991986_ 859378577423401	0.79989	0.003	FN
arquivopessoa.net/ textos/1395	facebook.com/ 100000487133835_ 930172310342331	0.99674	0.006	FN
arquivopessoa.net/ textos/3551	facebook.com/ 414828158557652_ 814940655213065	0.94336	0.01	FN

arquivopessoa.net/ textos/1145	facebook.com/ 203555383084745_ 636880729752206	0.729587	0.002	FN
arquivopessoa.net/ textos/827	facebook.com/ 100002568849764_ 645065342255755	0.753502	0.008	FN
arquivopessoa.net/ textos/2375	facebook.com/ 100001699713824_ 712194308847220	0.680845	0.003	FN
arquivopessoa.net/ textos/3548	facebook.com/ 1550142265_ 10204196912953598	0.525469	0.003	TN

B.3 Mensagens do Twitter que foram classificadas como citação

Obra	Mensagem	Score	Tempo	Class
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 409762612725809152	1.61485	0.007	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 409765871603970048	1.17223	0.02	TP
arquivopessoa.net/ textos/1027	twitter.com/ tweet/status/ 410531134762934272	1.07258	0.042	TP
arquivopessoa.net/ textos/508	twitter.com/ tweet/status/ 410562221194768384	1.49205	0.008	TP
arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 411335824341762048	1.02847	0.016	TP
arquivopessoa.net/ textos/4480	twitter.com/ tweet/status/ 411352589863636993	1.16571	0.006	TP
arquivopessoa.net/ textos/2512	twitter.com/ tweet/status/ 413953361923559424	1.00552	0.014	TP
arquivopessoa.net/ textos/4056	twitter.com/ tweet/status/ 415987637120679936	1.77175	0.007	TP
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 416019641388240896	1.10023	0.006	TP

arquivopessoa.net/ textos/503	twitter.com/ tweet/status/ 417332745258676224	1.24417	0.005	TP
arquivopessoa.net/ textos/3466	twitter.com/ tweet/status/ 417358448250998784	1.53303	0.006	TP
arquivopessoa.net/ textos/3446	twitter.com/ tweet/status/ 419649956048142336	1.89229	0.009	TP
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 419856117339271168	1.14701	0.005	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 419972892290732032	1.61485	0.007	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 420006038382084096	1.71705	0.007	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 420146822158376961	1.61485	0.006	TP
arquivopessoa.net/ textos/1584	twitter.com/ tweet/status/ 421337547541729280	1.1182	0.007	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 426013172282437632	1.61485	0.034	TP
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 422074895296913409	1.65085	0.037	TP
arquivopessoa.net/ textos/1027	twitter.com/ tweet/status/ 422798450582097920	1.35832	0.02	TP
arquivopessoa.net/ textos/3993	twitter.com/ tweet/status/ 422823375116189696	1.2958	0.011	TP
arquivopessoa.net/ textos/1646	twitter.com/ tweet/status/ 423457394937430016	1.10002	0.004	TP
arquivopessoa.net/ textos/2806	twitter.com/ tweet/status/ 424196871138332672	1.69391	0.003	FP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 424221516608589824	1.08332	0.006	TP

arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 424322546440237056	1.55202	0.009	TP
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 424534665496899584	1.40916	0.011	TP
arquivopessoa.net/ textos/567	twitter.com/ tweet/status/ 426851298596515840	1.0265	0.014	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 427622606746165248	1.28445	0.009	TP
arquivopessoa.net/ textos/508	twitter.com/ tweet/status/ 428537646294384640	1.89713	0.014	TP
arquivopessoa.net/ textos/2180	twitter.com/ tweet/status/ 411618154503106560	1.14208	0.205	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 434775856385245185	1.61485	0.004	TP
arquivopessoa.net/ textos/503	twitter.com/ tweet/status/ 435092041844801536	1.54719	0.003	TP
arquivopessoa.net/ textos/986	twitter.com/ tweet/status/ 436535302103240704	1.03223	0.007	TP
arquivopessoa.net/ textos/1122	twitter.com/ tweet/status/ 437417923595231232	1.26117	0.008	TP
arquivopessoa.net/ textos/3844	twitter.com/ tweet/status/ 437614168762744832	1.899	0.004	TP
arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 437913065439633408	1.13905	0.005	TP
arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 437970280187957248	1.52971	0.004	TP
arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 440438553638875136	1.52971	0.008	TP
arquivopessoa.net/ textos/602	twitter.com/ tweet/status/ 440439511945084928	1.05862	0.009	TP

arquivopessoa.net/ textos/4056	twitter.com/ tweet/status/ 440944821717716992	1.94222	0.005	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 440468078481399808	1.1013	0.009	TP
arquivopessoa.net/ textos/4056	twitter.com/ tweet/status/ 440944821717716992	1.94222	0.005	TP
arquivopessoa.net/ textos/2052	twitter.com/ tweet/status/ 440951272212938752	1.64578	0.005	TP
arquivopessoa.net/ textos/2583	twitter.com/ tweet/status/ 441040281966632960	1.05896	0.008	TP
arquivopessoa.net/ textos/1709	twitter.com/ tweet/status/ 441318561840001024	1.00044	0.006	TP
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 441462098166812672	1.14701	0.005	TP
arquivopessoa.net/ textos/4234	twitter.com/ tweet/status/ 443466469301825536	1.61485	0.009	TP
arquivopessoa.net/ textos/4056	twitter.com/ tweet/status/ 443917246839394304	1.56343	0.008	TP
arquivopessoa.net/ textos/2224	twitter.com/ tweet/status/ 444149794442076160	1.01965	0.002	TP
arquivopessoa.net/ textos/3844	twitter.com/ tweet/status/ 452084618804862976	1.899	0.005	TP
arquivopessoa.net/ textos/3844	twitter.com/ tweet/status/ 452088383440162818	1.899	0.006	TP

B.4 Mensagens do Twitter que não foram classificadas como citação

Obra	Mensagem	Score	Tempo	Class
arquivopessoa.net/ textos/1036	twitter.com/ tweet/status/ 409704008861749249	0.808568	0.006	TN

arquivopessoa.net/ textos/2507	twitter.com/ tweet/status/ 410806187278733312	0.501495	0.007	FN
arquivopessoa.net/ textos/2507	twitter.com/ tweet/status/ 410816623789240320	0.501495	0.006	FN
arquivopessoa.net/ textos/2507	twitter.com/ tweet/status/ 410853917418065920	0.656819	0.006	FN
arquivopessoa.net/ textos/2507	twitter.com/ tweet/status/ 410939402631008256	0.501495	0.004	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 411210235181678592	0.621185	0.005	TN
arquivopessoa.net/ textos/1362	twitter.com/ tweet/status/ 412272961177935872	0.607292	0.01	TN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 414216112222380032	0.521274	0.005	TN
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 416018798286618624	0.911546	0.005	FN
arquivopessoa.net/ textos/1103	twitter.com/ tweet/status/ 416201497194405888	0.590228	0.008	TN
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 416275326239379456	0.513594	0.005	FN
arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 416275601713295361	0.513594	0.006	FN
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 416696696761233408	0.755025	0.008	FN
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 416699801766756352	0.755025	0.005	FN
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 416706200844386304	0.755025	0.005	FN
arquivopessoa.net/ textos/3364	twitter.com/ tweet/status/ 411552506410704896	0.554209	0.007	FN

arquivopessoa.net/ textos/2492	twitter.com/ tweet/status/ 419717649036099584	0.678971	0.009	FN
arquivopessoa.net/ textos/1145	twitter.com/ tweet/status/ 419818580763750400	0.708286	0.01	FN
arquivopessoa.net/ textos/1275	twitter.com/ tweet/status/ 417102097252098049	0.507151	0.025	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 420682857493708800	0.621185	0.005	TN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 420942874180460544	0.67745	0.004	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 421025640843051009	0.621185	0.003	TN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 421757705208881152	0.517624	0.007	TN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 423074237361176576	0.507508	0.008	FN
arquivopessoa.net/ textos/1979	twitter.com/ tweet/status/ 424275653639536640	0.849354	0.007	TN
arquivopessoa.net/ textos/4518	twitter.com/ tweet/status/ 431261086801993728	0.571157	0.01	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 427135047129247744	0.579409	0.004	TN
arquivopessoa.net/ textos/4468	twitter.com/ tweet/status/ 427564987804958720	0.666121	0.009	FN
arquivopessoa.net/ textos/2206	twitter.com/ tweet/status/ 428530503117840384	0.786995	0.012	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 428978514453204992	0.551271	0.006	TN
arquivopessoa.net/ textos/3936	twitter.com/ tweet/status/ 429070631393853440	0.533083	0.003	TN

arquivopessoa.net/ textos/1584	twitter.com/ tweet/status/ 429947524817104896	0.979136	0.004	FN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 431094575948517377	0.687662	0.006	FN
arquivopessoa.net/ textos/1584	twitter.com/ tweet/status/ 431440956772995073	0.827163	0.009	FN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 431537179831971840	0.881248	0.004	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 432248943800365056	0.621185	0.005	TN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 433497071161585664	0.737152	0.006	FN
arquivopessoa.net/ textos/3834	twitter.com/ tweet/status/ 434459164342255616	0.703984	0.003	TN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 434880635824390144	0.621185	0.002	TN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 435610359978663936	0.621979	0.003	TN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 436101041637957632	0.621185	0.003	TN
arquivopessoa.net/ textos/2507	twitter.com/ tweet/status/ 436158449664409600	0.501495	0.003	FN
arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 436871944722731008	0.621185	0.004	TN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 436962250805370880	0.67745	0.005	FN
arquivopessoa.net/ textos/550	twitter.com/ tweet/status/ 437620150498787328	0.82946	0.005	FN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 441257735938916352	0.542426	0.008	FN

arquivopessoa.net/ textos/2700	twitter.com/ tweet/status/ 441562257630445568	0.621185	0.003	TN
arquivopessoa.net/ textos/2375	twitter.com/ tweet/status/ 441864273477001216	0.680845	0.006	FN
arquivopessoa.net/ textos/1463	twitter.com/ tweet/status/ 443103322187513856	0.67745	0.009	FN
arquivopessoa.net/ textos/1497	twitter.com/ tweet/status/ 452189717636067328	0.598165	0.006	FN

Apêndice C

Links de referências a "O Mundo em Pessoa"

Fonte	Link
RTP	rtp.pt/icmblogs/rtp/dicasinternet/?k=Dicas-O-Mundo-em-Pessoa-Fernando-Pessoa.rtp&post=45047
SIC Notícias	videos.sapo.tl/FmGz02ZzPeGMM49nHjYp
TSF	tsf.pt/PaginaInicial/Vida/Interior.aspx?content_id=3281760
Público	publico.pt/tecnologia/noticia/site-mostra-citacoes-de-fernando-pessoa-nas-redes-sociais-1597681
Visão	visao.sapo.pt/sabe-qual-e-o-poema-de-fernando-pessoa-mais-citado-na-internet=f736469
Expresso	expresso.sapo.pt/tabacaria-e-o-poema-mais-citado-de-pessoa-na-internet=f815278
TVI	tvi24.iol.pt/503/tecnologia/fernando-pessoa-versos-poema-tvi24/1462350-4069.html
Jornal de Noticias	jn.pt/PaginaInicial/Interior.aspx?content_id=3281285
Destak	destak.pt/artigo/167361-tabacaria-e-o-poema-de-fernando-pessoa-mais-citado-na-internet-estudo
Canal Superior	informacao.canalsuperior.pt/sala-geek/15808
Lux	luxwoman.pt/portfolio/nao-sou-nada-nunca-serei-nada-nao-posso-querer-ser-nada/
Marketeer	marketeer.pt/2013/06/20/o-mundo-em-pessoa-revela-os-versos-mais-citados/
IOL	iol.pt/push/iol-push—tecnologia/fernando-pessoa-versos-poema-tvi24/1462350-6186.html
Folha de São Paulo	www1.folha.uol.com.br/ilustrada/2013/06/1298689-projeto-reune-os-versos-de-fernando-pessoa-mais-citados-na-internet.shtml
Globo	m.g1.globo.com/pop-arte/noticia/2013/06/projeto-reune-os-versos-de-fernando-pessoa-mais-citados-na-internet.html

Yahoo! Brasil	https://br.noticias.yahoo.com/projeto-reúne-versos-fernando-pessoa-citados-internet-200407445.html
Info exame	info.abril.com.br/noticias/internet/2013/06/projeto-reune-os-versos-de-fernando-pessoa-mais-citados-na-internet.shtml

Apêndice D

Tabela de *web services* do "Lusica"

Operação	Consulta	Retorno
GET	albuns.php/{fonte}/date/{data}/estilo/{estilo}/	retorna a informação relativa aos álbuns citados no mês {data} e ao estilo {estilo}(*string* com o nome do estilo musical) se a {fonte} t̄witter. Caso a {fonte} l̄astfm, são devolvidas as entidades mais faladas nesse mês e estilo.
GET	albuns.php/top/tipo/{tipo}/inicio/{data1}/fim/{data2}/comp/{data3}/comp_fim/{data4}/	retorna o top do tipo definido (artistas, álbuns ou musicas) mais citado dentro do intervalo data1 até data2 comparando com o top dentro do intervalo data3 até data4 para devolver as posições no top que subiram e desceram.
GET	albuns.php/top/tipo/{tipo}/inicio/{data}/fim/{data}/comp/{data}/comp_fim/{data}/procura/{procura}/	Devolve, no top do tipo definido (artistas, álbuns ou musicas) mais citado dentro do intervalo {data1} até {data2} comparando com o top dentro do intervalo {data3} até {data4}, a posição neste top do termo {procura} dependendo do tipo. Se {tipo}=artistas, {procura} é procurado no nome do artista, se {tipo} ālbuns no nome do álbum e na {tipo} m̄usicas o nome da musica.

Apêndice E

Avaliação de "Lusica"

E.1 Mensagens do Twitter que foram classificadas como citação

Tweet	Artista	Música citada	Score	tempo	Class
https://twitter.com/tweet/status/138586185139359744	Jorge Palma	Página Em Branco	1.29392	0.014	TP
https://twitter.com/tweet/status/152388099224256512	Jorge Palma	Encosta-te A Mim	1.16645	0.015	TP
https://twitter.com/tweet/status/158038153591136256	Buraka Som Sistema	(We Stay) Up All Night	1.6563	0.014	TP
https://twitter.com/tweet/status/161810169532133376	Dealema	Verdade Ou Consequência	1.17389	0.013	TP
https://twitter.com/tweet/status/163318300057669632	Tony Carreira	Adeus Ate Um Dia	1.29066	0.009	TP
https://twitter.com/tweet/status/163927197541875712	Buraka Som Sistema	(We Stay) Up All Night	1.6563	0.011	TP
https://twitter.com/tweet/status/165664392271241216	Ornatos Violeta	Pára-me Agora	1.21454	0.007	TP
https://twitter.com/tweet/status/176918799247884288	Buraka Som Sistema	(We Stay) Up All Night	1.6563	0.016	TP
https://twitter.com/tweet/status/200674819837149184	Mundo Cao	Ordena Que Te Ame	1.67457	0.006	TP

https://twitter.com/tweet/status/202164650903740417	Mundo Cao	Ordena Que Te Ame	1.67457	0.006	TP
https://twitter.com/tweet/status/208865043360661505	Mundo Cao	Ordena Que Te Ame	1.67457	0.012	TP
https://twitter.com/tweet/status/209048224890236928	Jorge Palma	Encosta-te A Mim	1.47901	0.015	TP
https://twitter.com/tweet/status/213772586478342144	Chico Buarque	Essa moça tá diferente	1.12783	0.011	TP
https://twitter.com/tweet/status/214765446740709377	Balla	Outro Futuro	1.30738	0.01	TP
https://twitter.com/tweet/status/216468141738442752	Da Weasel	A Palavra - Tema para Sassetti (feat. Bernardo Sassetti)	1.20362	0.02	TP
https://twitter.com/tweet/status/217463458201997312	Silence 4	Borrow	1.14764	0.041	TP
https://twitter.com/tweet/status/217917763711275008	Ana Moura	O Que Foi Que Aconteceu	1.10246	0.014	TP
https://twitter.com/tweet/status/218277564270116865	Balla	Outro Futuro	1.30738	0.008	TP
https://twitter.com/tweet/status/218485089200377856	Rui Veloso	As Regras da Sensatez	1.03872	0.012	TP
https://twitter.com/tweet/status/233053428325171200	Da Weasel	A Palavra - Tema para Sassetti (feat. Bernardo Sassetti)	1.20362	0.022	TP
https://twitter.com/tweet/status/233539122638237697	Da Weasel	A Palavra - Tema para Sassetti (feat. Bernardo Sassetti)	1.20362	0.028	TP
https://twitter.com/tweet/status/237004988256239616	Silence 4	Borrow	1.14764	0.016	TP

https://twitter.com/tweet/status/238813137837625344	Os Azeitonas	Anda Comigo Ver os Aviões	1.53329	0.015	TP
https://twitter.com/tweet/status/242564593111666688	Da Weasel	A Palavra - Tema para Sassetti (feat. Bernardo Sassetti)	1.20362	0.015	TP
https://twitter.com/tweet/status/242702495674159104	Ornatos Violeta	Ouvi Dizer	1.06776	0.008	TP
https://twitter.com/tweet/status/244058708764680192	Jose Mario Branco	Qual é a Tua, ó Meu	1.07986	0.018	TP
https://twitter.com/tweet/status/245859289976152064	Os Azeitonas	Anda Comigo Ver os Aviões	1.53329	0.008	TP
https://twitter.com/tweet/status/246658821563482112	David Fonseca	What Life Is For	1.99276	0.009	TP
https://twitter.com/tweet/status/254873707854565376	Aurea	Scratch My Back	1.27817	0.009	TP
https://twitter.com/tweet/status/260579303195033601	Aurea	Scratch My Back	1.27817	0.011	TP
https://twitter.com/tweet/status/265430515232944132	Virgem Suta	Maria Alice	1.20211	0.012	TP
https://twitter.com/tweet/status/269069491261034497	David Fonseca	What Life Is For	1.99276	0.009	TP
https://twitter.com/tweet/status/271520144315011072	Virgem Suta	Maria Alice	1.20211	0.011	TP
https://twitter.com/tweet/status/275771413506428928	David Fonseca	Under the Willow	1.24518	0.013	TP
https://twitter.com/tweet/status/276340630295302144	Ivete Sangalo	Eu nunca amei alguém como te amei	1.60424	0.017	TP
https://twitter.com/tweet/status/276610182140796928	Aurea	Scratch My Back	1.27817	0.011	TP

https://twitter.com/tweet/status/280944751266656256	Expensive Soul	Dou-Te Nada	1.13491	0.012	TP
https://twitter.com/tweet/status/286029246206455808	Aurea	Scratch My Back	1.27817	0.009	TP
https://twitter.com/tweet/status/305442738483101696	Pedro Abrunhosa	Fazer O Que Ainda Não Foi Feito	1.42024	0.025	TP
https://twitter.com/tweet/status/309465282517295104	David Fonseca	All That I Wanted	1.15724	0.007	TP
https://twitter.com/tweet/status/311221483655544832	Aurea	Start Over	1.14579	0.006	TP
https://twitter.com/tweet/status/311753751805452288	Aurea	Start Over	1.14579	0.006	TP
https://twitter.com/tweet/status/319852695093645312	Tony Carreira	Porque não que vens?	1.9851	0.014	TP
https://twitter.com/tweet/status/363909674833543168	Madredeus	Agora - Canção aos Novos	1.15111	0.24	TP
https://twitter.com/tweet/status/370914370878193664	Pedro Abrunhosa	Momento (Uma Espécie De Céu)	1.28226	0.027	TP
https://twitter.com/tweet/status/407471857797853184	Sergio Godinho	A Vida não Feita de Pequenos Nadas	1.7801	0.017	TP
https://twitter.com/tweet/status/415922259899867136	Ivete Sangalo	Quando a chuva passar	1.25202	0.01	TP
https://twitter.com/ggiestas/status/423838892370849792	Ornatos Violeta	Quero ser feliz também	1.04131	0.008	TP
https://twitter.com/tweet/status/452168861392322561	Silence 4	Homem de Princípios	1.34063	0.026	TP
https://twitter.com/tweet/status/453231286870487040	Mesa	Vício de Ti	1.00398	0.006	TP

E.2 Mensagens do Twitter que não foram classificadas como citação

Tweet	Artista	Música citada	Score	Tempo	Class
https://twitter.com/tweet/status/1240733479182090 24	Chico Buarque	O meu amor	0.532036	0.009	FN
https://twitter.com/tweet/status/1538901798815129 62	Deolinda	Quando janto em restaurantes	0.82581	0.055	FN
https://twitter.com/tweet/status/1542653900951511 04	Vitorino	Maria da Fonte	0.618494	0.006	TN
https://twitter.com/tweet/status/3176840104133550 08	Doce	O Barquinho Da Esperança	0.502199	0.023	TN
https://twitter.com/tweet/status/1658502341597757 45	Adriana Calcanhotto	Fico assim sem você	0.775087	0.021	FN
https://twitter.com/tweet/status/1662006684854599 69	Jorge Palma	Voo Nocturno	0.816273	0.02	FN
https://twitter.com/tweet/status/1948406249549987 84	Os Azeitonas	Queixa ao Cupido	0.702083	0.014	FN
https://twitter.com/tweet/status/2074539942248611 86	Os Azeitonas	Queixa ao Cupido	0.702083	0.014	FN
https://twitter.com/tweet/status/2129207283900784 67	Chico Buarque	Mulheres de Atenas	0.961549	0.009	FN
https://twitter.com/tweet/status/2217091974720430 08	Paulo Gonzo	Espelho (De Outra água)	0.59537	0.014	FN
https://twitter.com/tweet/status/2176729516994887 68	Os Azeitonas	Anda Comigo Ver os Aviões	0.563715	0.009	TN
https://twitter.com/tweet/status/2246891122575319 06	Roberta Sa	Samba de um minuto	0.759781	0.007	FN
https://twitter.com/tweet/status/2251843246291845 12	Michel Telo	Ai se eu te pego!	0.944742	0.015	FN

https://twitter.com/tweet/status/225740749533229058	Linda Martini	As Putas Dançam Slows	0.72443	0.01	FN
https://twitter.com/tweet/status/226520926110502912	Joao Gilberto	Este seu olhar	0.517696	0.008	FN
https://twitter.com/tweet/status/229569494509551618	Jorge Palma	Portugal, Portugal	0.986121	0.017	FN
https://twitter.com/tweet/status/231326769762955264	Antonio Zambujo	Nem às Paredes Confesso	0.848354	0.026	FN
https://twitter.com/tweet/status/236605919306326016	Ornatos Violeta	Chaga	0.625324	0.009	FN
https://twitter.com/tweet/status/238382016297566209	Censurados	Srs Políticos	0.657009	0.006	FN
https://twitter.com/tweet/status/240586742414073856	Pedro Abrunhosa	Se Eu Fosse Um Dia o Teu Olhar	0.785707	0.021	FN
https://twitter.com/tweet/status/243870457462415360	Rui Veloso	O Prometido é Devido	0.936527	0.014	FN
https://twitter.com/tweet/status/244970094663450626	Sam The Kid	À Procura Da Perfeita Repetição	0.709997	0.018	FN
https://twitter.com/tweet/status/254914471238180864	Rui Veloso	O Prometido é Devido	0.936527	0.018	FN
https://twitter.com/tweet/status/260433085647171584	Aurea	Busy (for me)	0.547548	0.009	FN
https://twitter.com/tweet/status/269184365261631488	Mamonas Assassinas	Sabão cra-cra (The Mad ku-ku) (à putanesca)	0.885072	0.018	TN
https://twitter.com/tweet/status/276079759111368704	Toranja	Lados Errados	0.643552	0.013	FN
https://twitter.com/tweet/status/284005242385874944	Expensive Soul	Eu não sei	0.594913	0.008	FN

https://twitter.com/tweet/status/285396924247461889	Elis Regina	Madalena	0.670593	0.009	FN
https://twitter.com/tweet/status/290501777781030912	Herois do Mar	Só Gosto De Ti	0.697277	0.007	FN
https://twitter.com/tweet/status/295667223048101889	Clara Nunes	Tristeza Pé No Chão	0.658854	0.02	FN
https://twitter.com/tweet/status/292336168014606337	Biquini Cavadao	É Dia de Comemorar	0.542883	0.014	TN
https://twitter.com/tweet/status/314475828589232129	Ala dos Namorados	Caçador de sóis	0.817293	0.009	FN
https://twitter.com/tweet/status/207842458065838080	David Fonseca	What Life Is For	0.525446	0.009	FN
https://twitter.com/tweet/status/324157032833888257	Biquini Cavadao	Quando eu te encontrar	0.62455	0.016	FN
https://twitter.com/tweet/status/299937876567797762	Natiruts	Sorri, sou rei	0.522988	0.008	FN
https://twitter.com/tweet/status/332811926272888832	Ney Matogrosso	Balada do Louco	0.694029	0.013	FN
https://twitter.com/tweet/status/334374998229798912	Natiruts	No mar	0.545796	0.009	TN
https://twitter.com/tweet/status/384150718120476672	Adriana Calcanhotto	Do fundo do meu coração	0.553798	0.007	FN
https://twitter.com/tweet/status/350017549821419520	Paulo Gonzo	Call Girl	0.631933	0.019	FN
https://twitter.com/tweet/status/359148908133351424	Sara Tavares	Quando dás um pouco mai	0.51544	0.023	FN
https://twitter.com/tweet/status/384289498580127745	Linda Martini	Febril (Tanto Mar)	0.650572	0.012	FN
https://twitter.com/tweet/status/389101780589498368	Michel Telo	Se Tudo Fosse Fácil	0.66032	0.015	FN

https://twitter.com/tweet/status/398877054164553728	Ena Pa 2000	Vida De Cão	0.739412	0.015	FN
https://twitter.com/tweet/status/404027166477676544	Gilberto Gil	Vamos fugir	0.535174	0.009	FN
https://twitter.com/tweet/status/231366283613061120	Sergio Godinho	O elixir da eterna juventude	0.588143	0.006	FN
https://twitter.com/tweet/status/416367084550234112	Anselmo Ralph	Sem Ti	0.641816	0.011	FN
https://twitter.com/tweet/status/433562318115315712	Linda Martini	Dá-me a Tua Melhor Faca	0.563689	0.023	FN
https://twitter.com/tweet/status/443815405803819008	Natiruts	Sorri, sou rei	0.76554	0.011	FN
https://twitter.com/tweet/status/66180183496982528	Cristina Branco	Não há só tangos em Paris	0.62566	0.02	FN
https://twitter.com/tweet/status/78455424222900224	Maria Bethania	Volta por cima	0.891653	0.014	FN

Apêndice F

Links de referências a “Lusica”

Fonte	Link
TVI (minuto 29.40)	tvi.iol.pt/programa/jornal-da-uma/30/videos/128740/video/14122707/1
RTP (minuto 4.38)	rtp.pt/play/p827/e150480/tecnet
Tek	tek.sapo.pt/noticias/computadores/codebits_vii_o_cromossoma_xx_esta_a_crescer_n_1377688.html
PT	telecom.pt/InternetResource/PTSite/PT/Canais/Media/DestaquesHP/Destaques_2014/7_edicao_SAPO_Codebits.htm

Bibliografia

Rajendra Akerkar, Costin Bădică, and Dumitru Dan Burdescu. Desiderata for research in web intelligence, mining and semantics. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 0. ACM, 2012.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

Matko Boanjak, Eduardo Oliveira, José Martins, Eduarda Mendes Rodrigues, and Luís Sarmento. Twitterecho: a distributed focused crawler to support open research with twitter data. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1233–1240. ACM, 2012.

Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubicrawler: A scalable fully distributed web crawler. *Software: Practice and Experience*, 34(8):711–726, 2004.

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

Sergey Brin, Lawrence Page, and . The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

Soumen Chakrabarti. *Mining the Web: Discovering knowledge from hypertext data*. Morgan Kaufmann, 2003.

James Clark, Steve DeRose, et al. XML path language (xpath). *W3C recommendation*, 16, 1999.

Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana. Unraveling the web services web: an introduction to soap, wsdl, and uddi. *IEEE Internet computing*, 6(2):86–93, 2002.

David L Olson Dursun Delen. Advanced data mining techniques, 2008.

William B Frakes and Ricardo Baeza-Yates. Information retrieval: data structures and algorithms. 1992.

Otis Gospodnetic, Erik Hatcher, and . *Lucene*. Manning, 2005.

Jiawei Han, Micheline Kamber, and . *Data Mining, Southeast Asia Edition: Concepts and Techniques*. Morgan kaufmann, 2006.

Allan Heydon, Marc Najork, and . Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1):15–68, 2000.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

Nelson Leite, Hélder Caixinha, and Fernando Ramos. Proposta de uma aplicação web para monitorização do impacto de notícias nas redes sociais facebook e twitter. *Revista Comunicando, vol. 2*, 2013.

Bing Liu. *Web data mining*. Springer, 2007.

Sanjay Kumar Madria, Sourav S Bhowmick, W-K Ng, and Ee-Peng Lim. Research issues in web data mining. In *DataWarehousing and Knowledge Discovery*, pages 303–312. Springer, 1999.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

Jussi Myllymaki. Effective web data extraction with standard xml technologies. *Computer Networks*, 39(5):635–644, 2002.

Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, and Clemente Izurieta. Comparison of json and xml data interchange formats: A case study. *Caine*, 2009:157–162, 2009.

Eduardo Jorge Silva Leite de Oliveira. Twitterecho: crawler focado distribuído para a twittosfera portuguesa. Master's thesis, FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, 2010.

Leonard Richardson, Sam Ruby, and . *RESTful web services*. O'Reilly Media, Inc., 2008.

C Romaní, Hugo Pardo Kuklinski, and . Planeta web 2.0: Inteligencia colectiva o medios fast food. *Cadernos de Pesquisa*, 39(137), 2009.

J Savoy. Ir multilingual resources at unine, 2011.

Uwe Schindler, Benny Bräuer, and Michael Diepenbroek. Data information service based on open archives initiative protocols and apache lucene. 2007.

Sheila Tejada, Craig A Knoblock, and Steven Minton. Learning object identification rules for information integration. *Information Systems*, 26(8):607–633, 2001.

Guanhua Wang. Improving data transmission in web applications via the translation between xml and json. In *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, pages 182–185. IEEE, 2011.

Yan Wang. Web mining and knowledge discovery of usage patterns. *CS748T Project (Part I) Feb*, 2000.

Wouter Weerkamp, Simon Carter, and Manos Tsagkias. How people use twitter in different languages. *Proceedings of the Web Science*, 2011.