FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

TwitterEcho: *crawler* focado distribuído para a Twittosfera portuguesa

Eduardo Jorge Silva Leite de Oliveira

Mestrado Integrado em Engenharia Informática e Computação

Orientadora: Maria Eduarda Silva Mendes Rodrigues (Professora Auxiliar Convidada)

Co-orientador: Luís António Diniz Fernandes de Morais Sarmento (Assistente Convidado)

20 de Junho de 2010

TwitterEcho: *crawler* focado distribuído para a Twittosfera portuguesa

Eduardo Jorge Silva Leite de Oliveira

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: Jaime dos Santos Cardoso (Professor Auxiliar)
Vogal Externo: Sérgio Guilherme Aleixo de Matos (Equiparado a Investigador Auxiliar)
Orientador: Maria Eduarda Silva Mendes Rodrigues (Professora Auxiliar Convidada)

---

11 de Julho de 2010

Resumo

A *Web* disponibiliza aplicações que permitem aos utilizadores comunicarem entre si. Estas comunicações acontecem por exemplo em redes sociais como Facebook, Twitter, entre outras. A dimensão das redes sociais transforma-as num excelente veículo de *marketing*, propagação de informação, produção de notícias, realização de estudos de opinião, entre outros. A presente dissertação foca-se no Twitter que além de possuir a componente de rede social é também um serviço de *microblogging*, onde se publicam milhões de mensagens (*tweets*) por dia.

O problema que se coloca é como recolher dados da comunidade portuguesa presente no Twitter. É necessário ultrapassar as restrições e limites impostos pelo Twitter à recolha de dados. A recolha tem que ser contínua e todos os dados preservados. De todo o universo de utilizadores do Twitter é necessário identificar automaticamente os utilizadores portugueses.

A abordagem escolhida foi a implementação de um sistema distribuído e modular, denominado de TwitterEcho. A implementação de um sistema distribuído, permite que o ponto de estrangulamento (*bottleneck*) deixe de ser os limites impostos pelo Twitter, passando a ser o número de clientes que o servidor suporta. Com o objectivo de identificar os utilizadores portugueses, o TwitterEcho efectua a análise do perfil dos utilizadores e conteúdo dos *tweets*.

De 27 de Abril a 17 de Junho de 2011, o TwitterEcho adicionou aproximadamente 65 mil utilizadores, recolheu 2,8 milhões de *tweets*, cerca de 215 mil listas de seguidores e 130 mil listas de amigos. As avaliações efectuadas demonstraram bons resultados. Conseguindo 90.8% de classificações correctas de utilizadores portugueses e 99,5% de precisão na identificação de utilizadores não portugueses. A análise de linguagem possui uma precisão de 97,4% e abrangência de 65,9% na identificação da língua portuguesa de Portugal. Recolheu em média 75,2% dos *tweets* publicados pelos utilizadores.

O TwitterEcho permite a obtenção de dados em larga escala da comunidade portuguesa presente no Twitter. É um sistema complexo e inovador, um contributo valioso que torna possível a realização de estudos profundos sobre a comunidade portuguesa no Twitter. Devido ao interesse neste sistema, na presente data já foram desenvolvidas várias aplicações com base nos dados recolhidos por este.

Abstract

The Web has brought applications that allow users to communicate with each other. It is well shown for example in social networks like Facebook, Twitter and others. The popularity of social networks transforms them in an excellent way of marketing, information diffusing, studies of opinion and others. This thesis focused on Twitter, which besides being a social network is a microblogging service, with millions of short messages (called tweets) posted every day.

The problem addressed is collecting data of the Portuguese community in Twitter. It is necessary to overcome the restrictions and limitations imposed by Twitter to collect data. The process has to be continuous and all the data has to be maintained. It is necessary to identify Portuguese users among the whole community of Twitter users all over the world.

The approach chosen was the implementation of a distributed and modular system, named TwitterEcho. The implementation of the distributed system allows to collect data beyond the point allowed by Twitter. This way the bottleneck becomes the number of clients that the server supports. To identify Portuguese users, TwitterEcho analyzes user profiles and the textual content of tweets.

From 27th April to 17th June 2011, TwitterEcho collected about 65 thousand users, 2.8 million of tweets, 215 thousand followers lists and 130 thousand friends lists. The evaluations showed good results. Getting 90.8% correct classifications of Portuguese users and 99.5% precision in identifying non-Portuguese users. The language analysis has an precision of 97.4% and recall of 65.9% in the identification of the Portuguese language from Portugal. TwitterEcho collected on average 75.2% tweets posted by users every day.

TwitterEcho allows collecting large-scale data of the Portuguese community on Twitter. It is a complex and innovative system which is able to make a valuable contribution for in-depth studies of the Portuguese community on Twitter. Due to the capabilities in this system, several applications have already been developed based on data collected by TwitterEcho.

Agradecimentos

À Professora Eduarda Mendes Rodrigues pelo tempo dedicado a orientar esta dissertação.

Ao Professor Luís Sarmento pela orientação, principalmente na parte técnica, e pela rápida disponibilização dos recursos necessários para desenvolver esta dissertação.

Ao Eng. Gustavo Laboreiro pela discussão de ideias e pelo empenho demonstrado no acompanhamento da minha dissertação.

Ao José Martins pela colaboração no desenvolvimento desta dissertação.

A Monika Sobkowiak pela ajuda na formatação do presente documento.

Eduardo Jorge Silva Leite de Oliveira

Conteúdo

CONTEÚDO

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

Apache	Servidor Web de HTTP
API	Application Programming Interface
BFS	Breadth-first search
Bing	Motor de pesquisa da Microsoft
Crawler	Aplicação que percorre de forma metódica algo para recolher dados
Facebook	Rede social de grande dimensão na Web
GNIP	Guh'nip The Social Media API
Google	Motor de pesquisa
Hotmail	Serviço de e-mail gratuito da Microsoft
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
Levis	Empresa de produção e venda maioritariamente de calças de ganga
LOL	Laugh Out Loud
Mass media	Meios de comunicação em massa (televisão, rádio, imprensa, etc.)
Microblogging	Forma de publicação de textos curtos.
MySQL	Sistema de gestão de base de dados
N-gram	Segmentos de um texto
Ray-Ban	Empresa maioritariamente de produção de óculos de sol
REACTION	Retrieval, Extraction and Aggregation Computing Technology for Integrating and Organizing News
SQL	Structured Query Language
TwitPic	Serviço de partilha de fotos e vídeos no Twitter
Twitter	Serviço de microblogging. Possui também a componente de rede social.
Twittosfera	Termo usado para denominar tudo o que pertence ao universo do Twitter
URL	Uniform Resource Locator
Whitelist	Lista de entidades aprovadas
YouTube	Serviço de partilha de vídeos na Web

Capítulo 1

Introdução

1.1 Contexto/Enquadramento

A presente dissertação proposta pelos docentes Eduarda Mendes Rodrigues e Luís Sarmento enquadra-se no contexto do projecto *REACTION*, cujo principal intuito é desenvolver ferramentas de pesquisa, produção e apresentação na área do jornalismo computacional.

O constante fluxo de informação no século XXI obriga a novas práticas jornalísticas para monitorizar, interpretar e sumarizar notícias de forma eficiente. As notícias já não são simplesmente produzidas e consumidas, mas antes evoluem ao longo do tempo com a interacção entre as agências de notícias e o público [XLD11].

A visão acima descrita encontra-se na origem do trabalho jornalístico computacional. Esta ideia remonta há algumas décadas, no entanto, só presentemente se compreendeu a necessidade da existência de tecnologias avançadas de prospecção de dados.

Ao longo desta dissertação foco-me na prospecção de dados do Twitter, serviço de *microblogging* que permite aos utilizadores publicar textos curtos (ver figura 1.2). Possui também a componente de rede social permitindo a existência de relações entre utilizadores.

A escolha de elaborar a dissertação no contexto *Web*, mais especificamente em *microblogging* e redes sociais deve-se à importância em diversas áreas, como por exemplo:

1.1.1 Marketing viral

Marketing ou publicidade viral consiste num conjunto de técnicas que procuram usar redes sociais pré-existentes de forma a aumentar exponencialmente o reconhecimento e prestígio de uma marca. Isto pode ser conseguido pela passagem de palavra em redes sociais ou promoção através de vídeos, e-books, entre outros.

Uma das primeiras organizações a compreender a importância de marketing viral e a tirar proveito com grande sucesso foi a Hotmail. Colocou publicidade à própria, em todos os e-mails enviados pelos seus utilizadores [Wil]. A Hotmail é um caso de sucesso possuindo mais de 355 milhões de contas de e-mail [Mic].

Em Maio de 2007, foi publicado no sistema de vídeos on-line YouTube um vídeo de um indivíduo que atira um par de óculos de sol Ray-Ban a outro. Este apanha-o de diversas formas: num carro em movimento, dentro de um elevador, numa ponte, entre outros. Este vídeo foi lançado por um utilizador anónimo do YouTube e teve milhões de visualizações. Josh Warner, presidente da empresa The Feed Company, a posteriori explicou como promoveu este vídeo e que se trata de publicidade à Ray-Ban [Dou]. Em Maio do 2008, um vídeo semelhante ao anterior foi submetido no YouTube. Em vez de óculos de sol são usadas calças de ganga Levis. Erica Archambault, relações públicas da Levis, confirmou que a empresa esteve envolvida na criação do vídeo [Wor08].

A Old Spice possui vídeos no YouTube com milhões de visualizações[1]. Sobre estes anúncios existe uma página[2] na rede social Facebook, em que é possível efectuar questões e estas serem respondidas. A página teve um sucesso enorme, com mais de um milhão de pessoas a marcarem esta com "Gosto", esta acção pode ser executada por utilizadores do Facebook e usualmente encontra-se associada à manifestação de interesse.

Além destes exemplos, existem outras campanhas de marketing similares com grande sucesso. As organizações já se aperceberam que as redes sociais e toda a *Web* são um excelente veículo de propagação de informação. Apesar de já muito ter sido feito, devido à dimensão das redes sociais, ainda existe muito por explorar, tornando-se motivador trabalhar nesta área.

1.1.2 Produção de notícias

As redes sociais são uma fonte de informação valiosa e de produção de notícias. Existem várias notícias que foram primeiro reveladas nas redes sociais e só depois nos *mass media*.

Em 2009, surgiram publicações no Twitter reportando a queda de um avião em Nova Iorque, no rio Hudson, quinze minutos antes de surgir a notícia nos *mass media*. O primeiro *tweet* encontrado é da autoria de Jim Hanrahan, quatro minutos após o avião cair, escreveu "I just watched a plane crash into the hudson riv [sic] in manhattan". Outro utilizador do Twitter fotografou o avião e submeteu a imagem no TwitPic, serviço que permite a partilha de fotos no Twitter. Esta imagem foi disseminada rapidamente, causando uma interrupção dos serviços do TwitPic devido ao aumento do número de utilizadores. Este episódio vem provar o efeito bola de neve presente nas redes sociais e a sua excelência na propagação de informação [Bea09].

[1] http://www.youtube.com/watch?v=owGykVbfgUE
[2] http://www.facebook.com/OldSpice

Em 30 de Abril de 2011 o presidente dos Estados Unidos da América Barack Obama declarou que Bin Laden tinha sido morto por tropas americanas [BCM11]. A BBC noticiou que Sohaib Athar escreveu vários *tweets* quase em tempo real sobre o "ataque" a Bin Laden: "Helicopter hovering above Abbottabad at 1AM (is a rare event).", "A huge window shaking bang here in Abbottabad Cantt. I hope its not the start of something nasty :-S", entre outros [Tel11].

1.1.3 Poder político

Em Março de 2008, quando o governo da Malásia perdeu muitos lugares nas eleições, o anterior primeiro-ministro disse "We ... lost the Internet War... We made the biggest mistake in thinking that it was not important ... We tought that the newspapers, the print media, the television was supposed to be important, but the young people were looking at SMS and blog." [AMW10].

Em Dezembro de 2002, Trent Lott renunciou ao cargo de senador após ter sido criticado, por comentários inflamados que proferiu duas semanas antes. Apesar de este evento ter sido coberto pelos *media*, só uma semana depois foi dada relevância aos comentários racistas. Isto aconteceu, porque nesse período de tempo o assunto foi amplamente debatido na blogoesfera não o deixando cair no esquecimento [DF04].

O Twitter teve um papel importante na revolução na Tunísia, existem *tweets* a avisar da localização de atiradores furtivos, organizar protestos, pedir doações de sangue, entre outros [Car11].

No Egipto, Wael Ghonim um dos heróis de protesto de revolução diz: "This revolution started . . . in June 2010 when hundreds of thousands of Egyptians started collaborating content,", "We would post a video on Facebook that would be shared by 60,000 people on their walls within a few hours. I've always said that if you want to liberate a society, just give them the Internet.". Na figura 1.1 é possível ver na praça do Cairo de Tharir escrito no chão "Nós somos os homens do Facebook" [Gha11].

Devido à importância das redes sociais e *microblogging*, o objectivo da presente dissertação é criar um sistema que permita recolher dados de utilizadores portugueses presentes no Twitter. Esses dados são: informações gerais, mensagens e rede social. Foco-me na comunidade portuguesa devido a ser do interesse do Sapo Labs [3], mas este sistema deve ser escalável para outras comunidades de dimensão superior. Denominei este sistema como TwitterEcho.

[3] http://labs.sapo.pt/

Figura 1.1: Praça de Tharir no Cairo. Os protestantes escreveram: "Nós somos os homens do Facebook" [Gha11].

1.2 Breve introdução ao Twitter

Na página *about* do Twitter, este é descrito como uma rede de informação, em tempo real, que liga os utilizadores às últimas informações sobre tópicos que estes achem interessantes. A essência do Twitter consiste nos *tweets* que são pequenos pedaços de informação (máximo 140 caracteres). Os *tweets* também são denominados *status*, ou seja, o estado do utilizador. A página inicial do Twitter, após autenticação, permite publicar e visualizar *tweets* recentes de seguidores do utilizador autenticado (figura 1.2).

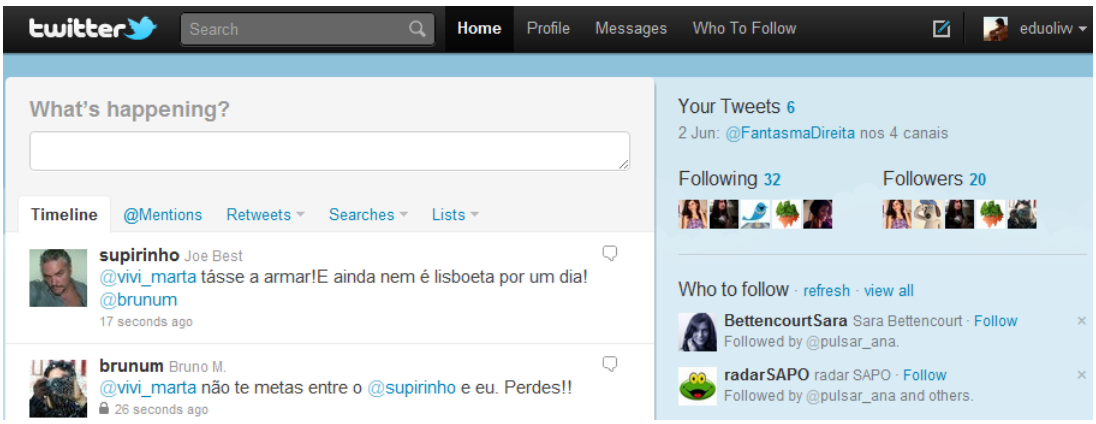


Figura 1.2: Página inicial do Twitter.

De seguida apresento um pequeno glossário de termos associados ao Twitter.

1.2.1 Histórico/*Timeline*

O histórico de um utilizador autenticado contém os *tweets* mais recentes dos utilizadores que este segue, ver figura 1.2.

1.2.2 Marcadores/*Hashtags*

O símbolo # permite marcar as palavras-chave ou tópicos de um *tweet*. Este símbolo no Twitter é denominado por *hashtag*. A existência de marcadores permite pesquisar *tweets* referentes a um determinado tópico [hcb]. Um exemplo do uso de marcadores ocorreu nas eleições legislativas portuguesas de 5 de Junho de 2011. O canal de televisão RTP incentivou utilizadores do Twitter a usarem o marcador *#e2011pt* nos seus *tweets* referentes à eleição (figura 1.3). Alguns destes apareceram na RTP durante o programa de análise das eleições na noite de 5 Junho.

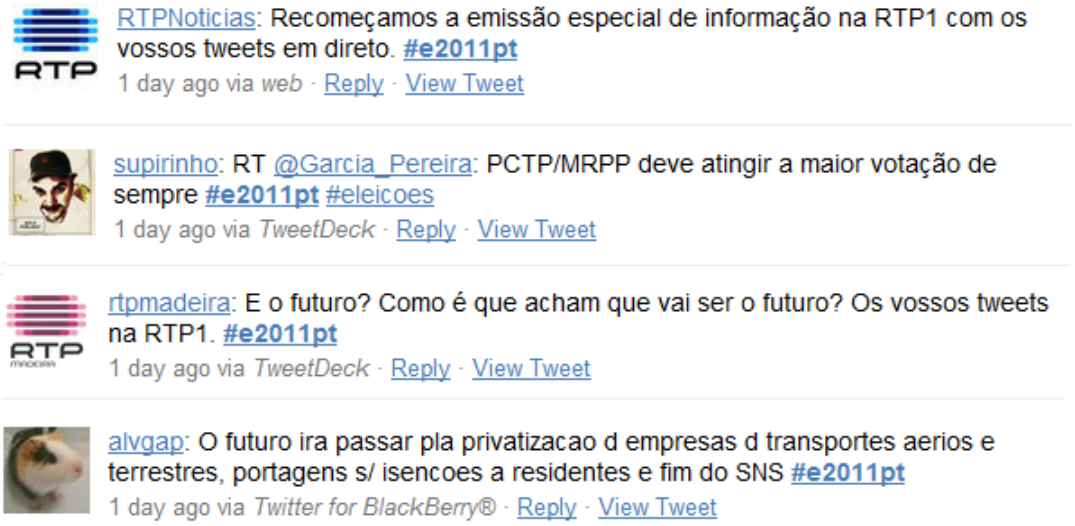


Figura 1.3: Exemplo de uso do marcador *#e2011pt*.

1.2.3 Menções/*Mentions*

Um utilizador pode mencionar outro. Este pode visualizar o *tweet* em que foi mencionado usando a secção *@Mentions* (ver figura 1.2). Uma menção é representada por *@nome* em qualquer parte do *tweet*. Um caso particular de menções é a resposta, é um *tweet* iniciado pelo nome do utilizador a quem este é dirigido [hcd]. A figura 1.4 é um exemplo de uma menção do utilizador *Publico* e a figura 1.5 uma resposta ao utilizador *FantasmaDireita*.

1.2.4 Seguidores/*Followers*

Utilizadores que seguem uma conta recebendo actualizações dos *tweets* desta. Esta relação não é necessariamente mútua, um utilizador pode seguir outro utilizador e este não o seguir [hca].

Figura 1.4: O utilizador *0vigia* menciona o utilizador *Publico*.

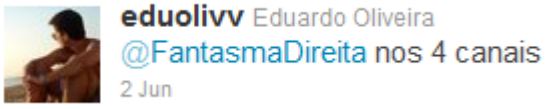


Figura 1.5: Resposta ao utilizador *FantasmaDireita*.

1.2.5 Amigos/*Friends*

Os amigos de um utilizador no Twitter são os utilizadores que este segue.

1.2.6 *Retweets*

Mecanismo que permite difundir um *tweet*. Um utilizador vê um *tweet* que lhe interessa e pode partilhá-lo com todos os seus seguidores, fazendo *retweet* deste [hcc], exemplo na seguinte figura:

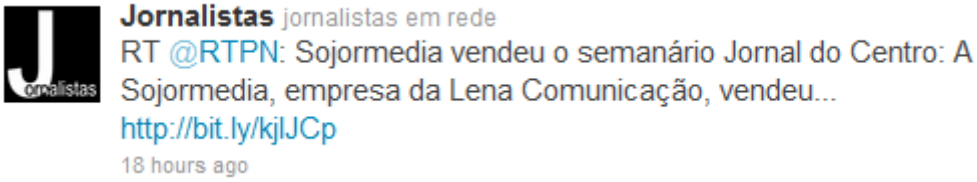


Figura 1.6: O utilizador Jornalistas faz *retweet* de um *tweet* do utilizador *RTPN*.

1.3 Motivação

Recolher dados do Twitter permite realizar importantes estudos e caracterizações. A minha principal motivação na realização desta dissertação é tornar possível o estudo da comunidade portuguesa no Twitter.

Algumas das caracterizações que foram efectuadas por diversos autores e que podem ser replicadas para a comunidade portuguesa:

1.3.1 Identificação de comunidades

Wang [Wan10] escolheu vinte utilizadores da *timeline* (secção A.1.1) do Twitter, as relações destes (seguidores e amigos) encontram-se representadas na seguinte figura:

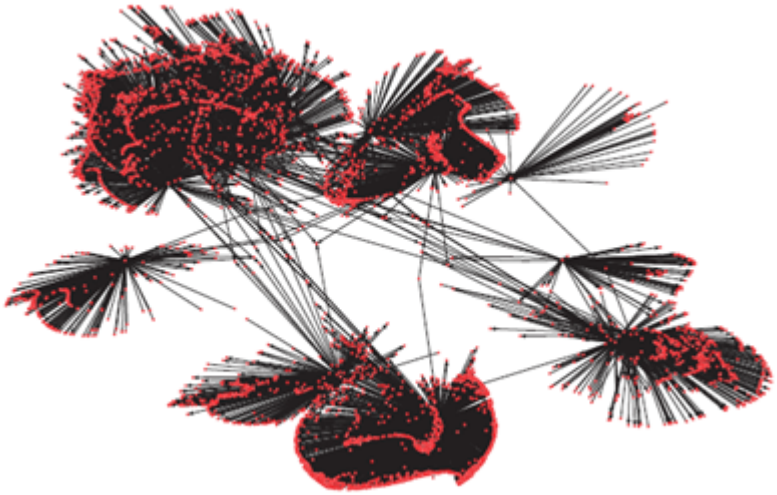


Figura 1.7: Relações entre utilizadores (seguidores e amigos) [Wan10].

Huberman et al. [HRW08] estudaram a importância de encontrar a verdadeira rede de amigos. Consideraram apenas as relações entre utilizadores em que existiam pelo menos duas respostas directas, ou seja, existia comunicação. Desta forma transformaram a rede da esquerda na mais simples à direita:

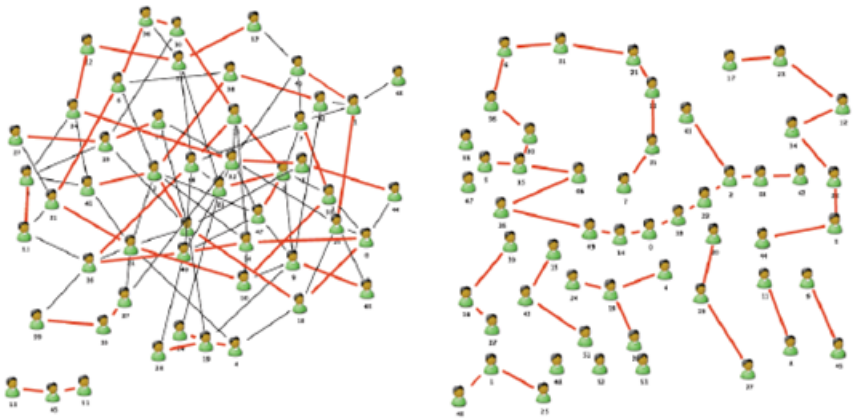


Figura 1.8: Simplificação da rede de amigos, mantendo apenas as ligações entre utilizadores em que existem pelo menos duas respostas directas [HRW08].

Estes estudos, efectuados por Wang [Wan10] e Huberman et al. [HRW08], são exemplos de identificação de comunidades. Existem outras divisões em comunidades que po-

dem ser efectuadas, como por exemplo a divisão por tópicos abordados. Na presente dissertação foco-me em identificar a comunidade portuguesa. Esta identificação possibilita a caracterização desta comunidade como um todo, porém, dividir esta em novos subconjuntos possibilita a realização de estudos mais pormenorizados.

1.3.2 Relações e *tweets*

Krishnamurthy et al. [KGA08] estudaram a relação entre o número de amigos e de seguidores. Esta caracterização encontra-se representada na figura 1.9. O eixo x representa o número de amigos e o eixo y o número de seguidores. Os utilizadores encontram-se divididos de acordo com o número de *tweets*: no percentual 99 possuem 1727 ou mais *tweets*, no percentual 90 possuem 964 ou mais *tweets* e os restantes denominados como *all* menos de 964 *tweets*.

Identificaram três grupos de utilizadores. O primeiro aparece como linhas verticais no lado esquerdo da figura 1.9. Estes utilizadores possuem um número de seguidores muito superior ao número de amigos e são regra geral muito activos (número elevado de *tweets*), indiciando que são difusores de informação tais como: estações de rádio, jornais e outros *media*. O segundo aparece em torno de y=x, ou seja, possui reciprocidade típica das redes sociais. Existe um terceiro grupo, em torno de x=7000, cujos utilizadores seguem muitos utilizadores. Seguir muitos utilizadores é um comportamento típico de *bots*. *Bot* é o diminutivo de *robot*, é uma aplicação informática concebida para simular acções humanas. No contexto do Twitter usualmente um *bot* pretende difundir informação, com esse objectivo segue o maior número de utilizadores possíveis na esperança que estes os sigam para depois propagar informação.

Esta caracterização é um passo significativo na identificação de diferentes perfis de utilizadores. As aplicações são variadas, por exemplo, caso se pretenda aferir o sentimento de utilizadores do Twitter relativamente a uma figura pública com grande impacto nos *media*. Se forem tidos em conta todos os utilizadores irão aparecer um número significativo de *tweets* com sentimento neutro. Isto deve-se à quantidade de jornais, rádios, entre outros órgãos da imprensa presentes no Twitter que geram milhares de *tweets* informativos. A identificação do perfil típico dos media permite excluir estes na aferição de sentimentos.

1.3.3 *Tweets* e eventos

Benevenuto et al. [BMRA10] demonstraram que eventos que possuem grande impacto na "vida real" reflectem-se no Twitter. Recolheram *tweets* contendo as palavras-chaves #musicmonday, "Susan Boyle", #susanboyle, "Michael Jackson", #michaeljackson e #mj, obtendo os resultados presentes na figura 1.10.

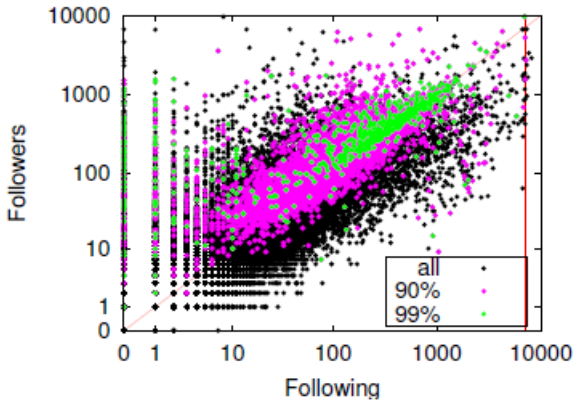
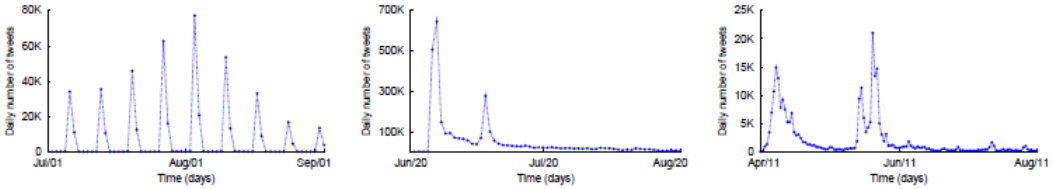


Figura 1.9: Relação entre o número de amigos e seguidores. O eixo x representa o número de amigos e o eixo y o número de seguidores. O percentual 99 refere-se a utilizadores que possuem 1727 ou mais *tweets*, percentual 90 utilizadores que possuem 964 ou mais *tweets* e os restantes com menos de 964 *tweets* são denominados *all* [BMRA10].

Topic	Period	Keywords	Tweets	Users
#musicmonday	Dec 8,2008—Sep 24,2010	#musicmonday	745,972	183,659
Boyle	April 10—Sep 24,2010	"Susan Boyle", #susanboyle	264,520	146,172
Jackson	Jun 25—Sep 24,2010	"Michael Jackson", #michaeljackson, #mj	3,184,488	1,232,865

Table 1: Summary information of three events considered to construct the labeled collection



(a) #musicmonday (b) Michael Jackson (c) Susan Boyle

Figura 1.10: Resultados da pesquisa efectuada por Benevenuto et al. [BMRA10] sobre a repercussão de eventos da "vida real" no Twitter.

O Music Monday possui picos a cada segunda-feira como esperado. O primeiro pico na figura 1.10 b) coincide com a morte de Michael Jackson em 25 de Junho de 2009 que teve um grande impacto na *media* [TMZ09a], o segundo com o funeral a 7 de Julho [TMZ09b]. Susan Boyle é uma cantora escocesa que se tornou famosa devido à participação no concurso de televisão "Britain's Got Talent" em 11 de Abril de 2009 [You11]. O segundo pico na figura 1.10 c) está relacionado com a final do concurso.

Na presente dissertação efectuo um estudo similar, com foco em programas televisivos portugueses. Demonstro uma correlação entre o número de referências e audiências televisivas. Este tema é abordado em detalhe no apêndice B.

1.3.4 Localização

Java et al. [JSFT07] com recurso à Yahoo Geocoding API [Yah] transformou as localizações presentes no perfil dos utilizadores do Twitter em coordenadas e gerou o seguinte mapa:

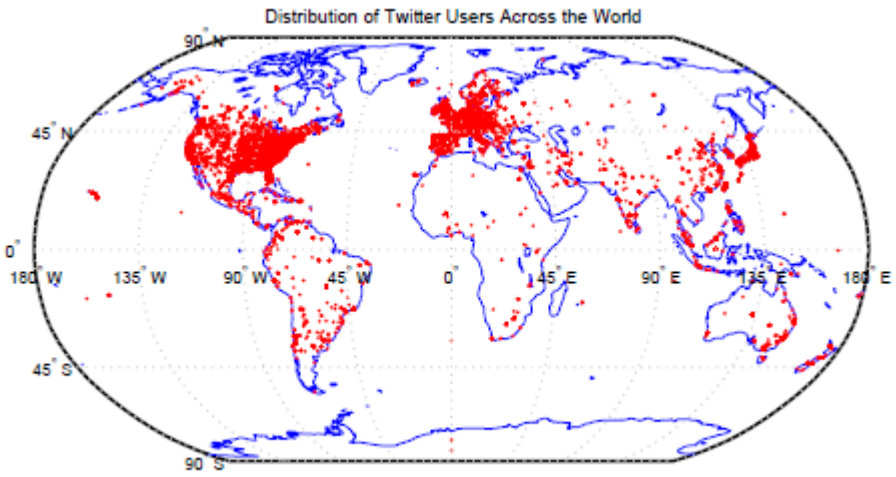


Figura 1.11: Localização dos utilizadores do Twitter recolhidos por Java et al. [JSFT07].

Este processo pode ser replicado para Portugal. Isto permite identificar as áreas do país em que o Twitter é mais usado. A análise de *tweets* sobre um dado acontecimento, ocorrido numa cidade com pouca expressão no Twitter possui um valor inferior a um acontecimento ocorrido numa cidade com muitos utilizadores presentes no Twitter, este estudo permite detectar esses casos. No jornalismo é usual a análise de opiniões por diferentes áreas, isto pode ser replicado para o Twitter.

1.3.5 Propagação de informação

Kwak et al. [KLPM10] estudaram o tempo de propagação de informação no Twitter. Na figura 1.12 está representado o tempo decorrido entre o *tweet* original e os *retweets* deste, metade dos *retweets* ocorre até uma hora após o *tweet* original, 75% em menos de um dia e curiosamente 10% um mês depois.

Kwak et al. [KLPM10] também estudaram o favoritismo na propagação de informação. Para cada utilizador **i** foi definido r**ij** como o número de *retweets* que este fez de *tweets* do utilizador **j**. Sendo Y(k,i) definido como:

$$Y(k,i) = \sum_{j=1}^{k} \left\{ \frac{\mid r_{ij} \mid}{\sum_{l=1}^{k} \mid r_{il} \mid} \right\}^2 \tag{1.1}$$

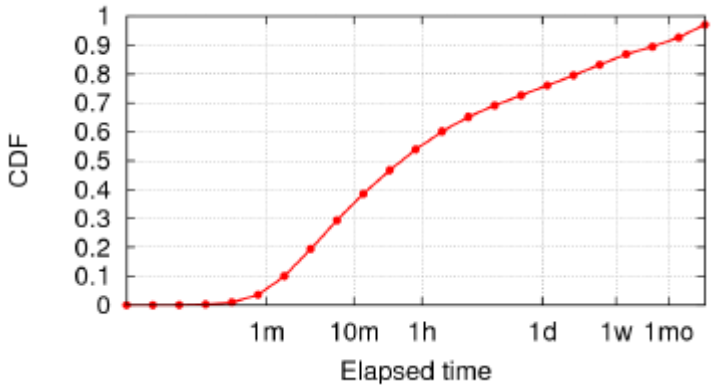


Figura 1.12: Tempo entre *retweets* e o *tweet* original [KLPM10].

Y(k) representa Y(k,1) calculado sobre todos os nós que possuem arestas (no sentido de saída). Um nó representa um utilizador e uma aresta representa um *retweet*. Quando a propagação de *tweets* ocorre de forma uniforme entre os seguidores, então $kY(k) \approx 1$. Se a maioria dos *retweets* ocorre num subconjunto de seguidores, então, então $kY(k) \approx k$. Na figura 1.13 é possível verificar uma correlação linear até 1000 seguidores. Esta correlação com k representa favoritismo na propagação de informação: utilizadores apenas propagam *tweets* de um pequeno número de pessoas e apenas um subconjunto de seguidores de um utilizador propagam *tweets* deste.

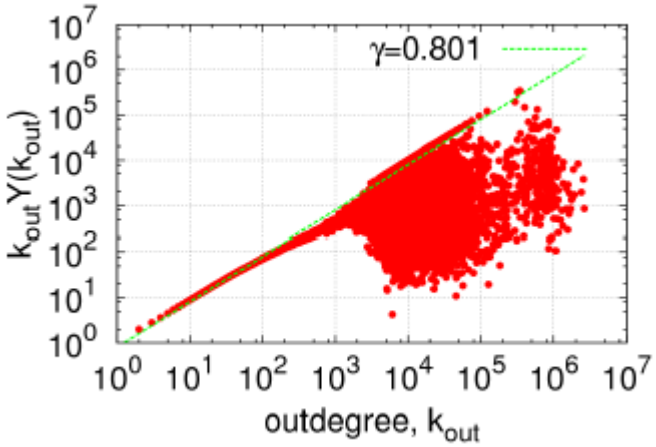


Figura 1.13: Estudo sobre o favoritismo na propagação de informação realizado por Kwak et al. [KLPM10].

Devido à descoberta da existência de favoritismo na propagação de informação por Kwak et al. [KLPM10] torna-se necessário identificar os utilizadores favoritos/influentes.

1.3.6 Identificação de utilizadores influentes

Existem pessoas que possuem uma extraordinária capacidade de influenciarem os outros na tomada de decisões, das mais pequenas, como por exemplo, onde jantar, sair, passear, até decisões mais importante como em quem votar. Uma pequena parte da população possui um grande poder sobre os restantes.

Keller e Berry [KB03] estudaram a população Americana concluindo que um em cada dez Americanos diz aos restantes nove como votar, onde comer e o que comprar. Os influentes são pessoas a quem os outros pedem conselhos sobre vários temas.

Estes conseguem prestar informações sobre assuntos tão diversos porque mantêm interesses em diversas áreas, sendo leitores assíduos de livros, jornais, revistas, entre outros. Estes indivíduos através da passagem de palavra possuem por vezes um poder superior à publicidade tradicional. É possível concluir que a temática de detecção de indivíduos influentes é de grande importância e possui diversas aplicações.

A identificação de utilizadores influentes na Twittoesfera tem sido alvo de diversos estudos [RGAH11] [CHBG] [SCHT07]. Possuir dados de utilizadores portugueses no Twitter permite identificar utilizadores portugueses influentes.

Cha. et al. [CHBG] recolheram dados de praticamente todo o Twitter e concluíram: o número de seguidores representa popularidade e não necessariamente influência (não se correlaciona com o número de *retweets* e menções). Os utilizadores podem ser influentes em diversas áreas. Os *retweets* são motivados pelo conteúdo do *tweet* e menções pela reputação do utilizador. Devido às conclusões anteriormente enunciadas quantificaram influência com base nos *retweets* e menções. Concluíram que usualmente a influência não é obtida espontaneamente e que é contínua no tempo como demonstrado na seguinte figura:

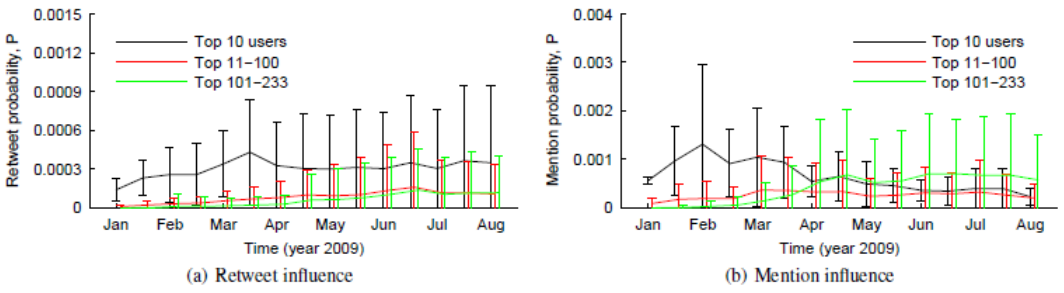


Figura 1.14: No gráfico a) encontra-se representada a probabilidade de ocorrência de *retweets* de um conjunto de utilizadores. O gráfico b) é similar ao a) tendo em conta a ocorrência de menções ao invés de *retweets* [CHBG].

1.3.7 Análise de sentimentos

Jansen et al. [JZSC09] seleccionaram cinquenta marcas e procuraram *tweets* mencionando-as. Usaram o *Summize*, ferramenta que permitia a pesquisa de *tweets* por palavras-chave e análise do sentimento geral de cada *tweet*, classificando-o numa escala de cinco pontos ("wretched, bad, so-so, swell and great"). Esta ferramenta foi adquirida pelo Twitter dando origem à Search API [Blo08] [Wil10]. Um total de 149472 *tweets* foi classificado da seguinte forma:

Sentiment by week	Occurrences	Percentage
Great	194	29.8%
Swell	200	30.8%
So-so	78	12.0%
Bad	102	15.7%
Wretched	42	6.5%
No Tweets	34	5.2%
Total	650	100.0%

Figura 1.15: Análise de sentimento referente a cinquenta marcas. O número de ocorrências refere-se à frequência absoluta, a percentagem refere-se à frequência relativa [JZSC09].

Os resultados presentes na figura 1.15 indicam que os utilizadores do Twitter usam-no para partilhar informações gerais e elaborar questões sobre marcas e produtos em adição à expressão de sentimentos. Uma pequena percentagem (12%) possui sentimento neutros. Os extremos são mais significativos, isto sugere que clientes com experiências extremamente positivas e negativas possuem maior probabilidade de fornecer informações face a utilizadores com experiências moderadas.

A análise de sentimentos referentes às marcas é de grande interesse para as empresas. A aferição de sentimentos pode ser efectuada noutros contextos que não marcas, um exemplo é o Twitómetro. Sistema que, tendo como base os dados recolhidos pelo TwitterEcho, efectuou a aferição do sentimento de portugueses presentes no Twitter em relação a líderes políticos. Abordo em maior detalhe este tema na secção 6.2.

1.4 Descrição do problema

A recolha é um problema complexo. Na realidade, é um conjunto de problemas distintos e complexos. Vou abordar de forma sucinta os principais desafios:

- Restrições do Twitter: este restringe o acesso aos dados limitando o número de pedidos. A política do Twitter tem vindo a mudar ao longo do tempo, dificultando a extracção de informação. Actualmente dados do Twitter estão disponíveis para venda no GNIP, denominada como The Social Media API. Na página *Web* do GNIP

encontra-se: "In November 2010, Gnip became the first authorized reseller of Twitter data" [GNI]. A venda de *tweets* levanta questões éticas que não se encontram no âmbito da presente dissertação debater, mas esclarece possivelmente um dos motivos porque o Twitter coloca tantos entraves à extracção de informação. Abordo em maior detalhe os limites na secção A.4.

- Recolha contínua: o sistema tem de ser capaz de se manter em funcionamento durante largos períodos de tempo (meses pelo menos). Tem portanto que ser robusto e capaz de recuperar de falhas. Uma falha grave, que impeça a recolha, representa a perda de dados numa janela temporal.
- Facilmente actualizável: quando se pretende proceder a alterações/melhorias não é aceitável ser necessário "desligar" o sistema para esse efeito (a recolha tem que ser contínua).
- Identificar utilizadores: identificar um utilizador como português não é uma tarefa trivial. Muitos portugueses escrevem em inglês. A língua portuguesa é usada noutros países com expressão maior no Twitter que Portugal (ex: Brasil). Os utilizadores do Twitter nem sempre preenchem os dados de localização (não é obrigatório o preenchimento).
- Dimensão de dados: o sistema não se limita a guardar o estado actual dos utilizadores, guarda toda a evolução das suas relações, *tweets* e estatísticas (número de *tweets*, seguidores e amigos). Estes dados têm de ser mantidos de forma estruturada para milhares de utilizadores.
- Fiabilidade: descobrir um erro, ainda que mínimo, que altere os dados recolhidos significaria que todos os trabalhos efectuados com os dados recolhidos pelo TwitterEcho se encontrariam comprometidos.

1.5 Estrutura do documento

O restante documento encontra-se divido nos seguintes capítulos:

2. Revisão bibliográfica: revisão de literatura importante para o desenvolvimento da presente dissertação.
3. Arquitectura do TwitterEcho: descrição alto nível do sistema.
4. Interface *Web*: descrição de algumas das funcionalidades do sistema *Web* de monitorização do TwitterEcho.
5. Avaliação: descrição e resultados dos métodos usados para avaliar o sistema.

6. Conclusões e Trabalho Futuro: conclusões, trabalho futuro e exemplos de aplicações.

Capítulo 2

Revisão Bibliográfica

2.1 Sistemas de *crawling*

2.1.1 *Web crawling*

Um *Web crawler* é uma aplicação que percorre a *Web* de forma metódica e automatizada.

Motores de pesquisa (Google, Bing, entre outros) percorrem a *Web* criando cópias de todas as páginas visitadas, para posterior processamento e indexação.

O trabalho efectuado e publicado na área é vasto. Em Junho de 1999 Heydon e Najork [HN99] lançaram o *Mercator*, um *crawler* cujo objectivo era indexar a *intranet* de organizações. Usaram-no também para percorrer a *Web*, em 8 dias efectuou cerca de 80 milhões de pedidos HTTP obtendo uma média de 122 documentos por segundo. Efectuaram a comparação da *performance* com o Google conseguindo números favoráveis.

A arquitectura base de *Web crawling* encontra-se descrita em vários livros e artigos. Manning e Raghavan [MR] descrevem as operações básicas de um *crawler*: começar com URL's conhecidos, obtê-los e analisa-los, extrair novos URL's, coloca-los numa fila, analisar os URL's na fila e repetir. O processo encontra-se representado na figura 2.1.

Na presente dissertação o objectivo é percorrer a Twittosfera ao invés da *Web*, no entanto, existem semelhanças, pensando em utilizadores em vez de páginas *Web* e interacções sociais (menções, seguidores e amigos) invés de hiperligações. Apesar da vasta literatura e das semelhanças, de agora em diante focar-me-ei na revisão de *crawler* específicos do Twitter. Isto deve-se à existência de diversos trabalhos sobre o Twitter, sendo que em muitos destes a primeira etapa consistiu na recolha de dados.

2.1.2 Twitter *crawling*

2.1.2.1 Gerais

Whitelist

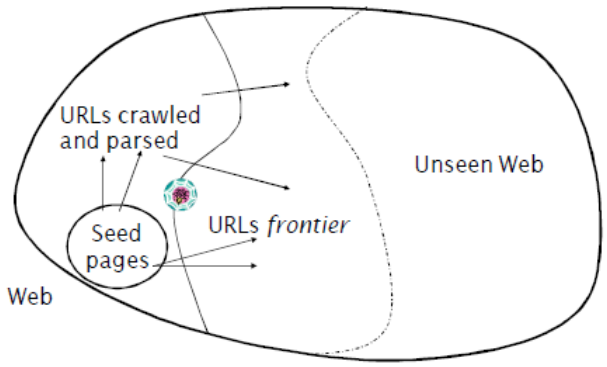


Figura 2.1: Representação do processo de *Web crawling* por Manning e Raghavan [MR].

O Twitter possui uma lista privilegiada (*whitelist*), as contas presentes na lista possuem limites muito acima do normal. Desta forma o acesso a whitelist permite a recolha de dados de forma mais abrangente.

Kwak et al. [KLPM10] recolheram dados do Twitter de 6 a 31 de Julho de 2009, possuindo *whitelist* (secção A.4) em vinte máquinas com diferentes IPs. Iniciaram uma pesquisa em largura (BFS) começando na Perez Hilton que possuía mais de um milhão de seguidores. Como forma de recolher dados de utilizadores que não estão interligados ao grafo principal foram efectuadas pesquisas na Search API por tópicos (secção A.2). Os tópicos escolhidos foram os mais populares, o Twitter disponibiliza o top 10 de tópicos, sempre que um novo tópico surgia eram efectuadas pesquisas durante 7 dias por este. Recolheram no total 4262 tópicos e seus *tweets*.

Kwak et al. [KLPM10] obtiveram 41,7 milhões de utilizadores. A metodologia usada na recolha de utilizadores não garante que todos foram recolhidos. Um exemplo de utilizadores que não são recolhidos é: um utilizador que não pertença ao grafo principal e não refira tópicos do top 10 e possua ligações apenas a utilizadores em situações similares.

Em Agosto do mesmo ano Benevenuto et al. [BMRA10] estudaram a detecção de *spam* no Twitter. O Twitter forneceu *whitelist* a 52 servidores, e estes requisitaram os dados de cada utilizador pelo identificador numérico de 0 a 80 milhões (maior identificador numérico presente nas listas de seguidores [MS]). Desta forma recolheram 54 milhões de utilizadores, 1900 milhões de relações e 1800 milhões de *tweets*.

Benevenuto et al. [BMRA10] praticamente no mesmo período temporal que Kwak et al. [KLPM10] conseguiram mais 12 milhões de utilizadores. Esta comparação indicia que pesquisas por tópicos pode não ser suficiente para apanhar os utilizadores que não estão ligados ao grafo principal.

Conseguir *whitelist* para 20 ou 52 servidores não é trivial. O acesso a *whitelist* possui grande importância quando se pretende recolher dados de praticamente todo o Twitter. O Twitter tem vindo a diminuir os acessos, na presente data, muitos dos pedidos são negados

(secção A.4). Irei portanto focar-me nos *crawlers* sem recurso a *whitelist*.

Sem *whitelist*

Um dos primeiros *crawlers* do Twitter é da autoria de Java et al. [JSFT07], usaram o método *Statuses/public_timeline*, que devolve vinte *tweets* recentes (secção A.1.1), a cada 30 segundos. Recolheram dados de 1 de Abril até 30 de Maio de 2007, aproximadamente, 1 milhão e 300 mil *tweets* e cerca de 76 mil utilizadores. Esta recolha de dados coincide com a fase embrionária do Twitter, em que este possuía uma dimensão muito inferior à actual. Na presente data o método usado por Java et al. [JSFT07] não é suficiente para recolher uma quantidade de dados significativa.

Wang [Wan10] recolheu dados do Twitter de 3 a 24 de Janeiro de 2010. Nesse período de tempo recolheu, aproximadamente, 25 mil utilizadores, meio milhão de *tweets* e 49 milhões de relações. Tal como Java et al. [JSFT07] recorreu ao método de *Statuses/public_timeline* (secção A.1.1). Para cada *tweet* recolhido foram extraídos os utilizadores mencionados. Para cada um destes, recorrendo a REST API, foi obtida a lista de seguidores e amigos. Foram obtidos mais *tweets* dos utilizadores acedendo directamente à página do Twitter destes. Wang [Wan10] usou um sistema distribuído para este efeito e refere que se limitou a 120 pedidos por hora e por máquina, devido ao limite de 150 pedidos por hora e por IP imposto pela API do Twitter.

Jansen e Lussier [JL10] pretendiam testar ferramentas de análise de redes desenvolvidas por estes. Construíram um *crawler*, usando a REST API, com o objectivo de recolher relações entre utilizadores. Começaram num utilizador e expandiram obtendo a rede dos amigos e seguidores deste, e assim por diante. Chegaram à conclusão que demoraria demasiado tempo a recolher uma quantidade significativa de dados devido ao limite de 150 chamadas por hora (pediram *whitelist* sem sucesso). A solução de recurso foi "povoar" a base de dados com utilizadores e relações falsas para poderem testar as ferramentas de análise de redes. Concluíram que se tivessem mais dados poderiam ter efectuado análises mais interessantes.

2.1.2.2 Focados

Pak e Paroubek [PP10] recolheram dados do Twitter e construíram um classificador de sentimentos capaz de determinar sentimentos positivos, negativos e neutros. Usaram a API do Twitter para recolher trezentos mil *tweets*, pesquisando por emoções positivas (“:-)”, “:)”, “=)”, “:D” etc.), por emoções negativas (“:-(”, “:(”, “=(”, “;(” etc.). Como forma de recolher *tweets* neutros recolheram *tweets* de 44 jornais.

Galuba et al. [GAC+10] estudaram a propagação de URL’s no Twitter durante 300h, com início em Setembro de 2009. Usaram a Search API para pesquisar por "http" (secção

A.2). Os *tweets* foram analisados, URL's e nomes de utilizadores extraídos. Para cada *tweet* foram pedidos os dados do autor e os utilizadores que este segue. O resultado final consistiu nos URLS's mencionados datados e o grafo social. Foram recolhidos 27 milhões de *tweets* e 15 milhões de URL's entre 2,7 milhões de utilizadores.

Lopes et al. [LZT+09] recolheram *tweets* contendo referências a doenças e localizações. A recolha foi efectuada recorrendo à Search API (secção A.2). Foram pesquisadas 89 doenças, em diversos países, no total 17615 pesquisas. O processo de recolha demorou entre 2 a 3 dias e inicia-se a cada semana. Caso fosse pretendido começar a verificar um maior número de países e doenças, o processo de recolha tornaria-se demasiado longo, a solução poderia passar pela implementação de um sistema distribuído. Actualmente o Twitter, disponibiliza a Streaming API (secção A.3) que tem como objectivo lidar com este tipo de situações, e poderia ser uma solução mais adequada para este problema em particular.

2.1.3 Conclusões

É possível retirar algumas ilações dos trabalhos revistos:

- O *crawling* do Twitter possui similaridades com *Web crawling*. É possível combinar o uso da API do Twitter com *Web crawling* [Wan10].
- É necessário possuir um conhecimento profundo das APIs e métodos do Twitter. Existem trabalhos revistos que poderiam sofrer uma melhoria com uma mudança de API ou método [GAC+10] [LZT+09].
- Quando a quantidade de dados a recolher é importante a opção por um sistema distribuído deve ser equacionada, esta opção foi tomada por diversos autores [KLPM10] [BMRA10] [Wan10].
- Uma alternativa ao uso de um sistema distribuído é o uso de várias contas por máquina para aumentar a quantidade de dados recolhida. Alguns dos trabalhos revistos poderiam ter beneficiado desse método [JL10].

2.2 Identificação de linguagem

O problema de identificação de linguagem a partir de uma amostra de texto foi abordado de diferentes formas:

2.2.1 Combinações únicas de letras

Churcher em 1994 [Dun94] [Kra] propôs um conjunto de combinações de letras específicas de cada língua. Um exemplo de combinações está presente na tabela 2.1.

Língua	Combinação
Dutch	"vnd"
English	"ery"
French	"eux"
Gaelic	"mh"
German	"der"
Italian	"cchi"
Portuguese	"seu"
Serbo croat	"lj"
Spanish	"ir"

Tabela 2.1: Exemplos de combinações de letras específicas de cada língua [Dun94].

Por si só este método não é suficiente para efectuar a identificação de uma língua, num texto de pequena dimensão as combinações únicas de letras podem não ocorrer. As combinações de letras podem ocorrer noutras línguas, por exemplo a referência de Montreux não significa que o texto é francês e discutir Pinocchio não significa que o texto é italiano. Este método é predecessor da identificação por N-grams.

2.2.2 Palavras comuns curtas

Grefenstette [Gre95] estudou a identificação de linguagem com palavras comuns curtas. Usou um corpus de notícias de jornais (Corpus from European Corpus Initiative), analisou o primeiro milhão de caracteres para cada língua dividindo estes em palavras pelo carácter espaço (tokenização). Foram apenas retidas palavras de cinco caracteres ou menos e com mais de três ocorrências. Foram recolhidas entre 980 a 2750 palavras comuns curtas para cada língua. Criou uma lista de probabilidades frequências de cada língua. A frequência de cada palavra foi transformada na probabilidade aproximada de ocorrência, dividindo a sua frequência pela soma de todas as frequências das palavras comuns pequenas retidas.

Dado um texto para análise, este é dividido e são retidas as palavras curtas. Para cada palavra é verificada para cada língua se esta aparece na lista de frequência, caso apareça é atribuída a frequência caso contrário uma frequência mínima. A classificação final é obtida como o produto das probabilidades de todas as palavras.

Grefenstette [Gre95] realizou testes com textos de diferentes dimensões, tendo o sistema que decidir entre 10 línguas diferentes. Os resultados estão presentes na figura 2.2.

O sistema não possui bons resultados na identificação de textos de pequena dimensão, tornando-se bastante preciso em textos de grande dimensão. Este abordagem não é facilmente aplicável em todas as línguas, por exemplo a divisão da língua chinesa em palavras não é linear.

	Number of Words in Sentence							
	1 or 2	3 – 5	6 – 10	11 – 15	16 – 20	21 – 30	31 – 50	more than
				Danish				
short	40.5	61.6	91.8	94.8	95.5	94.3	92.5	100.0
				Dutch				
short	47.1	84.2	98.5	99.2	99.5	99.6	99.9	100.0
				English				
short	52.6	87.7	97.3	99.8	99.9	100.0	99.9	100.0
				French				
short	30.8	81.8	96.0	97.2	99.8	100.0	100.0	100.0
				German				
short	23.1	71.6	89.6	98.2	99.8	100.0	100.0	100.0
				Italian				
short	16.7	65.0	96.9	99.8	100.0	100.0	99.9	100.0
				Norwegian				
short	87.5	97.4	99.2	99.8	99.9	100.0	100.0	100.0
				Portuguese				
short	51.1	88.9	98.2	99.7	99.9	99.9	100.0	100.0
				Spanish				
short	8.1	81.5	98.8	99.7	100.0	100.0	100.0	100.0

Figura 2.2: Precisão do algoritmo de identificação de línguas proposto por Grefenstette [Gre95].

2.2.3 N-grams

Um N-gram é uma parte de N tokens de um texto. Um token pode ser qualquer segmento, caracteres, palavras, etc. Neste contexto específico os N-Grams referem-se a caracteres. A principal vantagem do uso de N-grams é que erros ortográficos apenas afectam uma parte dos N-grams deixando os outros intactos [CT94] [Kra].

Uma das abordagens de maior sucesso foi introduzida por Cavnar e Trenkle [CT94]. Apesar de uma maior preocupação com a categorização de textos, descobriram que os seus métodos têm sucesso na identificação da língua.

A ideia base do uso de N-grams para a identificação de linguagem é que, invariavelmente, todas as línguas possuem palavras que ocorrem com maior frequência que outras. Uma das formas mais comuns de expressar esta ideia é denominada Zipf's Law:

> "The size of the r'th largest occurrence of the event is inversely proportional to it's rank r." [Kra]

Esta lei implica que existe sempre um conjunto de N-grams que domina uma língua em termos de frequência. A curva de frequência pode ser observada na figura 2.3.

A natureza da curvatura ajuda, porque implica que não é necessário ter em conta todos os N-grams.

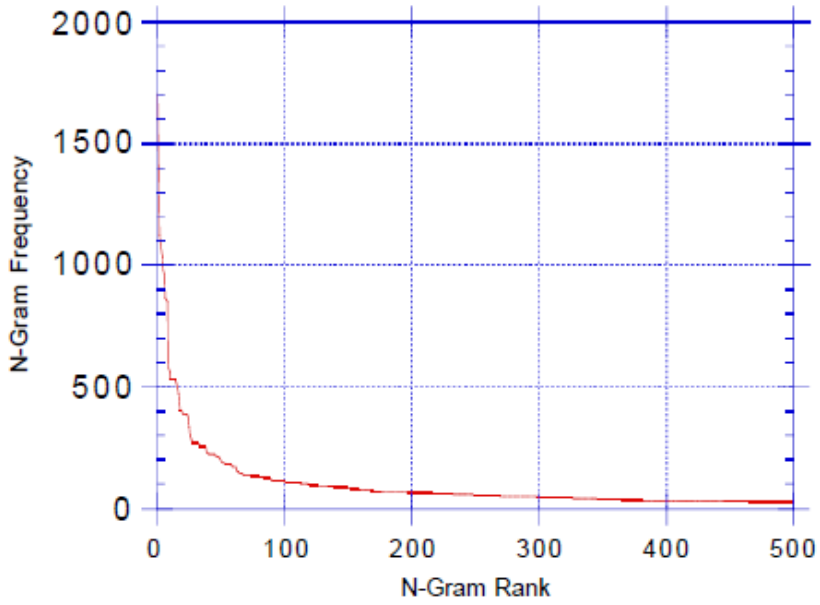


Figura 2.3: Curvatura que demonstra a existência de N-grams dominantes [CT94].

Cavnar e Trenkle [CT94] construíram perfis de amostras de 10KB até 200KB. Para cada amostra foram geradas as frequências dos N-grams. Para classificar um novo documento, o sistema cria o perfil deste, e compara-o com os outros perfis. O sistema classifica o documento como pertencendo à categoria que este mais se assemelha.

O método usado para gerar o perfil foi: separar o texto apenas em letras e apóstrofos (tudo o resto é descartado), gerar todos os N-grams para N=1 até 5, contagem de ocorrências de cada, ordenar do N-gram com maior número de ocorrências para o com menor número de ocorrências. A arquitectura do sistema está presente na figura 2.4.

Cavnar e Trenkle [CT94] retiraram algumas conclusões dos perfis criados:

- O top 300 N-grams estão usualmente relacionados com a língua, um texto de poesia ou sobre compiladores em Inglês possui muitos N-grams em comum nos seus perfis no top 300. No entanto um texto Francês em qualquer tópico possui uma distribuição diferente de qualquer texto em Inglês.
- Após os 300 N-grams surgem N-grams mais específicos do tema.

Estas observações foram maioritariamente retiradas de documentos de pequena dimensão. Cavnar e Trenkle [CT94] referem que em documentos de maior dimensão a mudança de N-grams específicos da linguagem para N-grams específicos do tema provavelmente ocorrerá mais tarde.

A comparação entre dois perfis é efectuada através da soma das distâncias, um exemplo de cálculo encontra-se presente na figura 2.5.

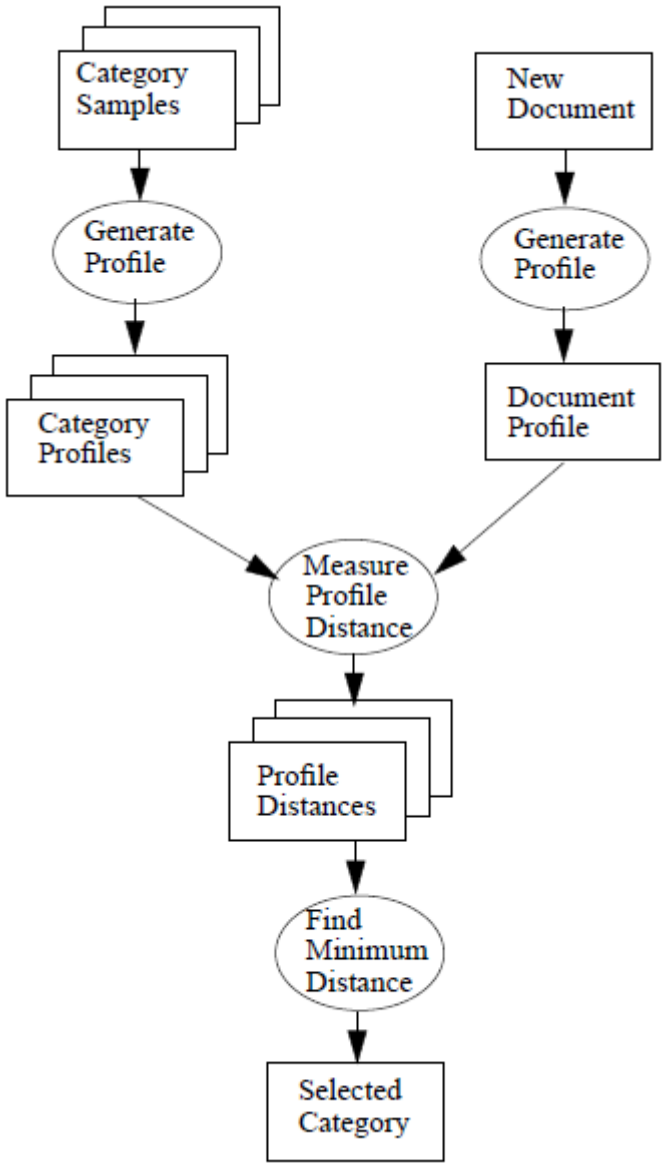


Figura 2.4: Arquitectura do classificador proposto por Cavnar e Trenkle [CT94].

Cavnar e Trenkle [CT94] realizaram os testes com 3713 amostras recolhidas de soc.culture newsgroup. Estes *newsgroups* eram usados para discussões sobre tópicos relevantes aos países em particular e usualmente na língua do país.

Os resultados encontram-se disponíveis na figura 2.6 e estão disponíveis em dois tamanhos de textos e quatro tamanhos de perfis usados.

Cavnar e Trenkle [CT94] concluíram que o classificador trabalha apenas ligeiramente melhor com textos mais longos, mas menos que o esperado. Para a maioria dos casos quanto maior o perfil melhores resultados de classificação. No entanto ocorreram anomalias (indicadas com asterisco na figura 2.6). Em análise posterior identificaram que

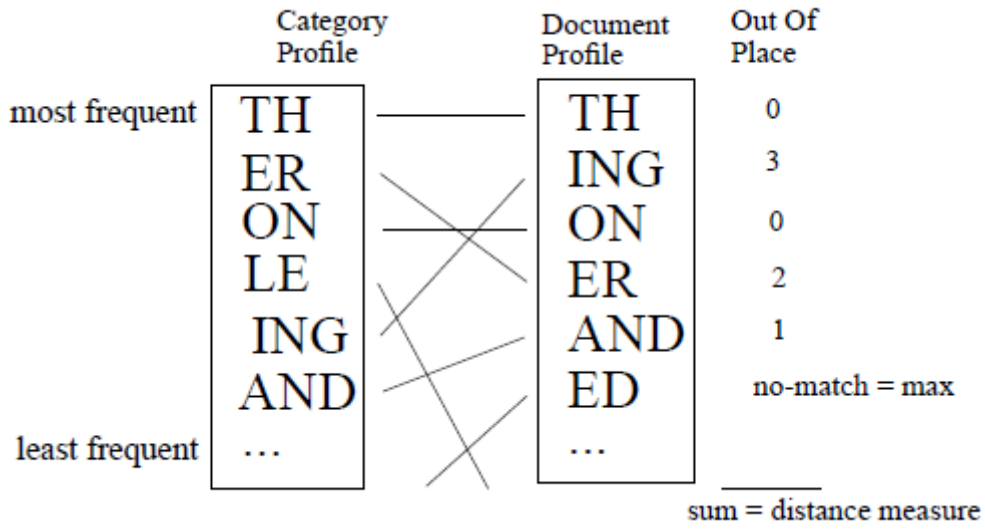


Figura 2.5: Exemplo de cálculo da distância [CT94].

alguns textos possuíam duas línguas. O sistema não possuía um mecanismo eficaz para lidar com este problema, sendo forçado a escolher entre dois perfis com distâncias similares do texto. Adicionar N-grams (perfil maior) pode colocar um dos perfis à frente do outro de uma forma difícil de prever.

O uso de N-grams tem sido alvo de vários estudos e aprofundamentos. John Prager [Pra99] modelou o problema de forma geométrica. Usou treze línguas europeias, comparando os resultados usando N-grams de diferentes tamanhos, palavras, palavras curtas (quatro caracteres ou menos) e combinações de diferentes métodos. Concluiu que a combinação de métodos permite um aumento da precisão, sendo que o melhor resultado é conseguido com 4-grams e palavras sem restrições de tamanho. Os resultados são apresentados na figura 2.7.

Ashmed et al. [ACT04] referem que Cavnar e Trenkle [CT94] mencionaram os resultados e o tamanho do conjunto de dados de treino, mas não a velocidade. O algoritmo proposto por Cavnar e Trenkle [CT94] requer ordenação, que é uma operação pesada. Ashmed et al. [ACT04] propõem um novo classificador usando um perfil de N-grams similar, mas que não requer ordenação. Ao invés de usar estatísticas de posicionamento, usa adição cumulativa de frequência. Esta abordagem possui resultados similares aos do algoritmo proposto por Cavnar e Trenkle [CT94] com ganhos de velocidade significativos.

Elworthy [Elw99] propõe que apenas sejam processados caracteres suficientes para atingir o grau de confiança pretendido. Desta forma é possível aumentar a velocidade, especialmente em textos longos.

Article Length	**≤ 300**	**≤ 300**	**≤ 300**	**≤ 300**	**> 300**	**> 300**	**> 300**	**> 300**
Profile Length	**100**	**200**	**300**	**400**	**100**	**200**	**300**	**400**
Newsgroup								
australia	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
brazil	70.0	80.0	90.0	90.0	91.3	91.3	95.6	95.7
britain	96.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
canada	100.0	100.0	100.0	100.0	100.0	*99.6	100.0	100.0
celtic	100.0	100.0	100.0	100.0	99.7	100.0	100.0	100.0
france	90.0	95.0	100.0	*95.0	99.6	99.6	*99.2	99.6
germany	100.0	100.0	100.0	100.0	98.9	100.0	100.0	100.0
italy	88.2	100.0	100.0	100.0	91.6	99.3	99.6	100.0
latinamerica	91.3	95.7	*91.3	95.7	97.5	100.0	*99.5	*99.0
mexico	90.6	100.0	100.0	100.0	94.8	99.1	100.0	*99.5
netherlands	92.3	96.2	96.2	96.2	96.2	99.0	100.0	100.0
poland	93.3	93.3	100.0	100.0	100.0	100.0	100.0	100.0
portugual	100.0	100.0	100.0	100.0	86.8	97.6	100.0	100.0
span	81.5	96.3	100.0	100.0	90.7	98.9	98.9	99.45
Overall	**92.9**	**97.6**	**98.6**	**98.3**	**97.2**	**99.5**	**99.8**	**99.8**

Figura 2.6: Precisão das avaliações do algoritmo proposto por Cavnar e Trenkle [CT94], o "Article Length" é dado em bytes e o "Profile Length" em número de N-grams.

	Chunk Size					
Feature-set	20	50	100	200	200	200
2-grams	68.8	86.2	93.5	97.7	98.8	100.0
3-grams	79.5	93.0	97.7	99.3	100.0	100.0
4-grams	83.6	94.3	98.2	99.6	99.9	100.0
5-grams	81.4	93.1	97.8	99.4	99.9	99.9
Words	69.7	86.6	94.7	98.1	99.9	100.0
SWords	61.3	81.5	92.1	97.1	99.6	100.0
SW+3grams	83.8	94.9	98.5	99.7	100.0	100.0
SW+4grams	84.9	95.3	98.6	99.7	99.9	100.0
W+4grams	85.4	95.6	98.7	99.7	99.9	100.0

Figura 2.7: Precisão dos vários métodos propostos por John Prager [Pra99]. Linhas correspondem ao método e colunas ao tamanho em bytes. Os valores são referentes à média das treze línguas.

2.2.4 Abordagens baseadas em compressão

Fitzgerald [Fit] sugere a seguinte experiência:

1. Comprimir dois ficheiros de dimensão significativa em duas línguas diferentes (anotar os tamanhos finais), serão denominados perfil A e perfil B.

2. Escolher uma amostra de uma das línguas e concatenar com ambos os ficheiros (perfil A + amostra e perfil B + amostra).

3. Compactar os ficheiros obtidos no ponto 2 e anotar os tamanhos finais.

4. Calcular a diferença dos valores obtidos no passo 3 pelos obtidos no passo 1.

A língua que causar uma menor diferença é provavelmente a língua da amostra. Isto pode ser explicado porque as técnicas de compressão representam as ocorrências mais frequentes com códigos de menor dimensão e línguas diferentes possuem frequências diferentes. Esta técnica foi usada com bons resultados por Teahan e Harper [TH] com base em modelos de compressão PPM (Prediction by Partial Macthing). Esta técnica possui uma vantagem face ao N-gram, não necessita de estabelecer os limites das palavras pois a compressão pode ser efectuada por conjuntos de caracteres.

2.2.5 Abordagem estatística

Dunning [Dun94] usa o modelo de Markov, este é um processo em que o próximo estado depende apenas do estado actual. A figura 2.8 possui um exemplo de modelo Markov de dois estados.

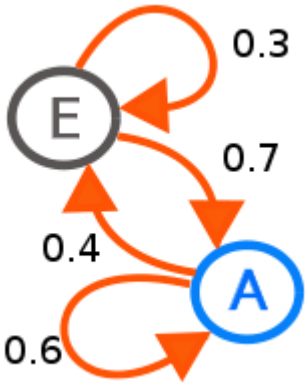


Figura 2.8: Exemplo de modelo Markov de dois estados. Os estados representa os uni gramas E e A.

Esta abordagem possui bons resultados em textos de pequenas e grandes dimensões. A abordagem proposta por Dunning [Dun94] ao contrário de N-grams, não necessita de pré-processamento dividindo o texto em palavras.

2.2.6 Conclusões

Em suma, a combinação de letras únicas não é forte o suficiente para identificação da língua. Palavras comuns curtas não são fiáveis, em textos de pequena dimensão, porque podem não ocorrer. N-gram, compressão e abordagem estatísticas produzem bons resultados em textos de pequena e grande dimensão. O método de N-gram necessita de pré-processamento para divisão em palavras não sendo universal (línguas asiáticas).

2.3 Sumário

No presente capítulo abordei sistemas de *crawling* e identificação de linguagem. A revisão de sistemas de crawling focou-se maioritariamente em *crawlers* do Twitter, sendo abordados *crawlers* com e sem recurso a *whitelist*, gerais e focados. Na identificação de linguagem abordei vários métodos: combinações únicas de letras, palavras comuns curtas, n-grams, abordagens baseadas em compressão e abordagem estatística.

Capítulo 3

Arquitectura do TwitterEcho

Definir a arquitectura deste projecto foi a fase mais importante deste. Na secção 1.4 descrevo os principais problemas cuja resolução é necessária para recolher dados do Twitter. A resolução destes influencia directamente a arquitectura do TwitterEcho. Os problemas e respectivas soluções adoptadas são:

- Restrições do Twitter: implementação de um sistema distribuído, desta forma o ponto de estrangulamento (*bottleneck*) deixa de ser os limites impostos pela API do Twitter mas sim quantos clientes o servidor suporta.
- Recolha contínua: a implementação de um sistema distribuído possui vantagens na recolha de dados contínua. Por exemplo se uma conta for bloqueada pelo Twitter apenas afecta um cliente, a recolha de dados prossegue com menos um cliente.
- Facilmente actualizável: o servidor é constituído por pequenos módulos (secção 3.4), com funções distintas. Estes são facilmente substituíveis por novas versões.
- Identificar utilizadores: criação de um módulo de análise de perfil (secção 3.4.3) e outro de identificação de linguagem (secção 3.4.5).
- Dimensão de dados: uso de MySQL, tendo em especial atenção os índices e velocidade de execução das instruções SQL (secção 3.2).
- Fiabilidade: desenvolvimento modular (secção 3.4) e testes exaustivos, na presente data o sistema recolhe dados há aproximadamente 45 dias.

O TwitterEcho consiste num sistema distribuído composto por um servidor que armazena dados e clientes que interagem com este e com o Twitter (figura 3.1). Um cliente interage com o servidor através de serviços, estes por sua vez efectuam operações na BD. Possui também diversos módulos que efectuam tratamento de dados de forma periódica. A arquitectura do TwitterEcho encontra-se representada em maior detalhe na figura 3.2.

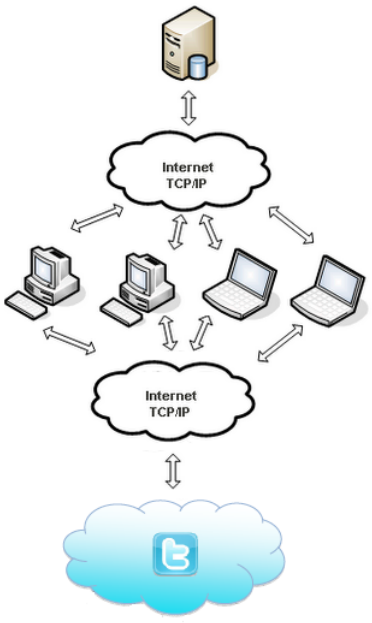


Figura 3.1: Servidor e clientes que comunicam entre si através da Internet.

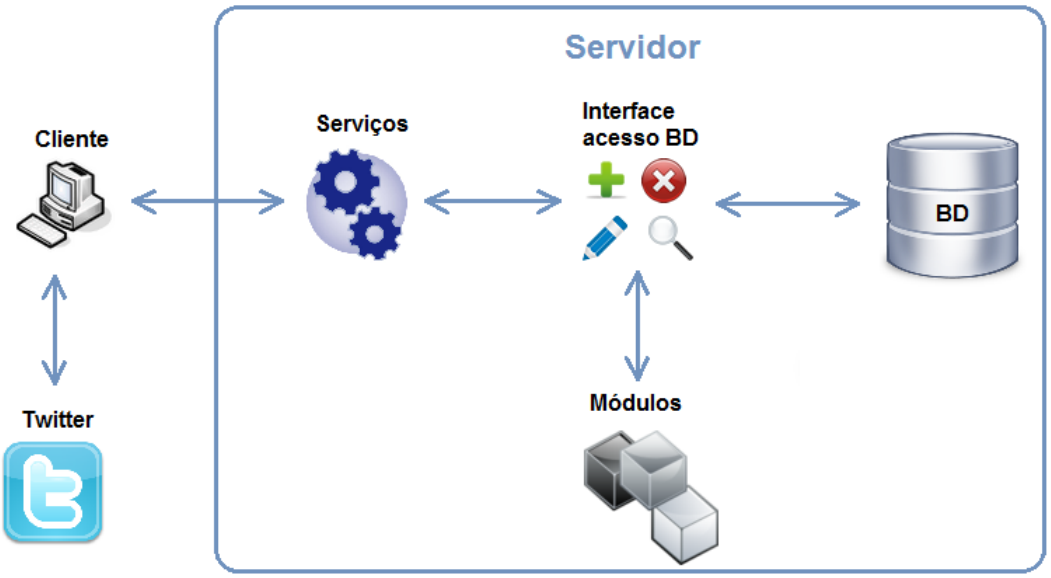


Figura 3.2: Arquitectura do TwitterEcho.

3.1 Clientes

Na figura 3.2 refiro cliente no singular, pois trata-se da arquitectura geral. O sistema é escalável para suportar múltiplos clientes. Por múltiplos entenda-se tanto em função como quantidade.

Actualmente o sistema possui dois clientes com funções distintas: recolha de *tweets* e relações entre utilizadores (amigos e seguidores). Os dados são enviados para o servidor no formato JSON[1].

Os clientes são da autoria do meu colega José Martins no âmbito da dissertação deste [Mar11]. No presente documento descrevo apenas o essencial para a compreensão do processo de recolha de dados.

3.1.1 Cliente lookup

A principal função deste cliente é a recolha de *tweets*. Para esse efeito usa o método *Users/lookup* da REST API do Twitter (mais informações na secção A.1.3). Além dos *tweets* envia para o servidor o perfil dos utilizadores e estatísticas (número de *tweets*, seguidores e amigos).

3.1.2 Cliente links

Recolhe as relações entre utilizadores. Para cada utilizador efectua os pedidos *Friends/ids*(ver secção A.1.4) e *Followers/ids*(ver secção A.1.5) à REST API do Twitter. Envia para o servidor a lista de seguidores e amigos.

3.2 Base de Dados

O TwitterEcho armazena um grande volume de dados (*gigabytes* e milhões de registos). De forma a permitir pesquisas e tratamento estatístico os dados são armazenados de forma estruturada. Usa o sistema de gestão de base de dados relacional MySQL.

A estrutura da BD é complexa, nesta secção explico de forma abreviada a informação presente nesta e o seu uso.

O servidor guarda as informações disponibilizadas pelos clientes: perfil dos utilizadores, *tweets*, estatísticas, lista de seguidores e lista de amigos.

Para tornar possível a detecção de falhas guarda: erros gerados por si ou pelos clientes, o último acesso de cada cliente aos serviços (*Ping*) e a evolução do tamanho das tabelas da BD.

Guarda também informação que permite o crescimento da BD e selecção de utilizadores portugueses. São guardados possíveis utilizadores, que são utilizadores de quem ainda não se possui informação suficiente para tomar a decisão de nacionalidade, ou foram considerados como não sendo portugueses. Para permitir a selecção de utilizadores são guardadas localidades portuguesas e não portuguesas, nomes não portugueses e *tweets* de utilizadores para identificação da língua.

[1] http://www.json.org

A estrutura da BD foi criada tendo em mente que esta iria armazenar grandes quantidades de dados. Os índices foram escolhidos cuidadosamente. A sua utilização foi privilegiada na implementação da interface de acesso à BD. Usei técnicas de *profiling* [Sch] para detectar e evitar o uso de instruções SQL lentas.

3.3 Serviços

Os serviços são responsáveis por coordenarem os clientes e receberem dados destes. Estes são: *get*, *put* e *put_error*. O *get* disponibiliza uma lista de utilizadores de quem se pretende dados, o *put* recebe dados e o *put_error* recebe erros. A figura 3.3 representa as interacções entre os vários agentes do sistema.

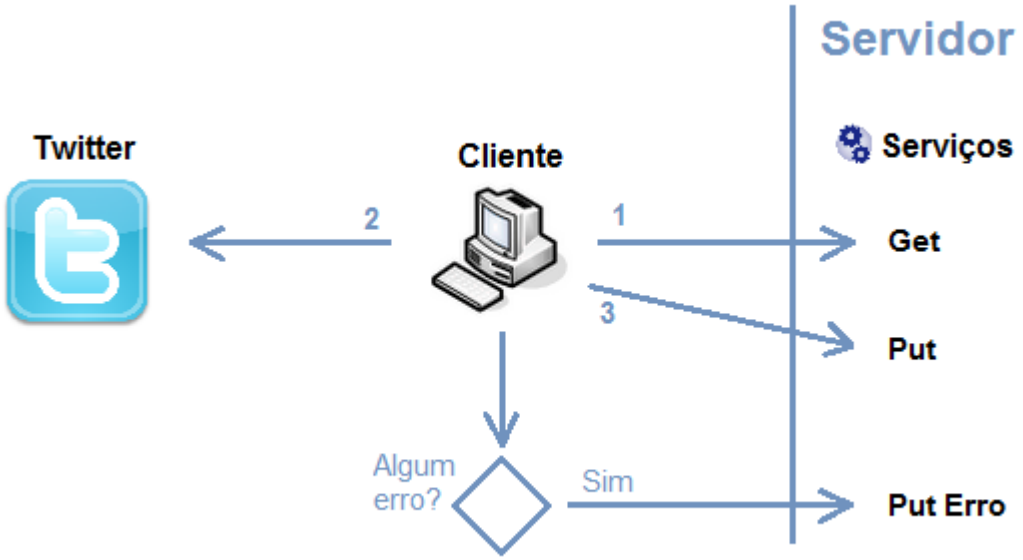


Figura 3.3: Interacções entre os agentes do sistema.

As acções são executadas na seguinte ordem:

1. O cliente acede ao serviço *get*, de forma a obter a lista de utilizadores de quem se pretende obter informações.
2. Efectua um pedido à API do Twitter referente aos utilizadores obtidos em 1.
3. Envia os dados obtidos em 2 para o serviço *put*.

Em qualquer das acções acima enunciadas o cliente, caso se aperceba de um erro do servidor, seu ou da API do Twitter, comunica o erro usando o serviço *put_error*. Neste caso volta ao ponto 1.

O serviço *get* utiliza um algoritmo de escalonamento para gerar uma lista de utilizadores de quem se pretende informações, ou seja, é responsável por coordenar os clientes. O serviço *put* recebe dados dos clientes e insere-os de forma estruturada na BD. Ao longo do presente capítulo abordo em maior detalhe o funcionamento de cada serviço.

3.3.1 Serviços lookup

Os serviços descritos na presente secção são responsáveis pela interacção entre clientes lookup e o servidor.

***Get* lookup**

Este serviço gera uma lista de utilizadores de quem o servidor pretende informações. Esta lista contém possíveis utilizadores (candidatos por analisar) e utilizadores marcados como portugueses. São seleccionados possíveis utilizadores de quem ainda não se possui dados do perfil. Esses dados são necessários para tomar a decisão de nacionalidade. Por sua vez, os utilizadores são escolhidos com base num algoritmo de escalonamento, explico este algoritmo em detalhe na secção 3.3.4. A sequência de acções pode ser observada na figura 3.4.

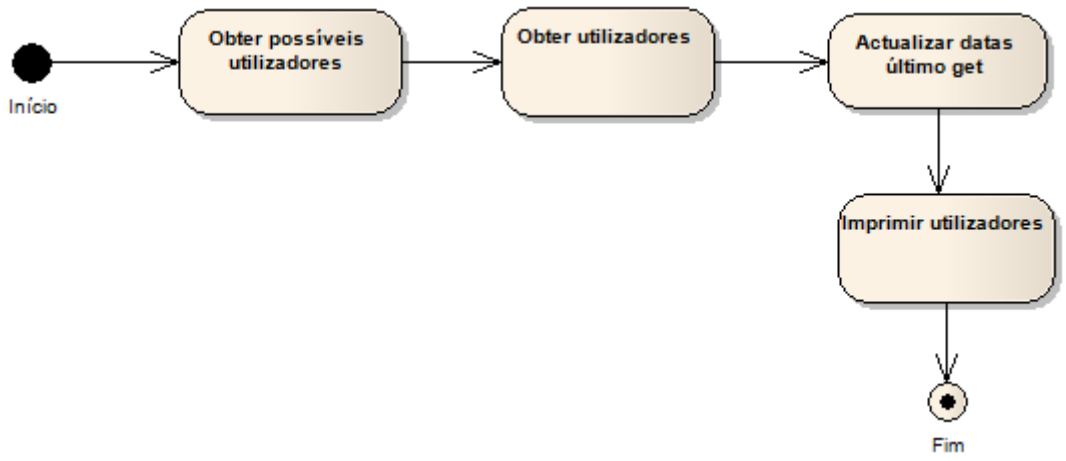


Figura 3.4: Acções do serviço *get* lookup.

***Put* lookup**

Este serviço recebe dados enviados pelos clientes lookup. De forma abreviada, este serviço executa um conjunto de verificações, actualiza informações de utilizadores, insere estatísticas, insere *tweets* e modifica prioridades. A sequência das acções pode ser observada na figura 3.5.

As maiorias das acções sobre a BD são acumuladas e apenas efectuadas no final do processamento (figura 3.5 "Realizar actualizações e inserções"). Esta opção deve-se à

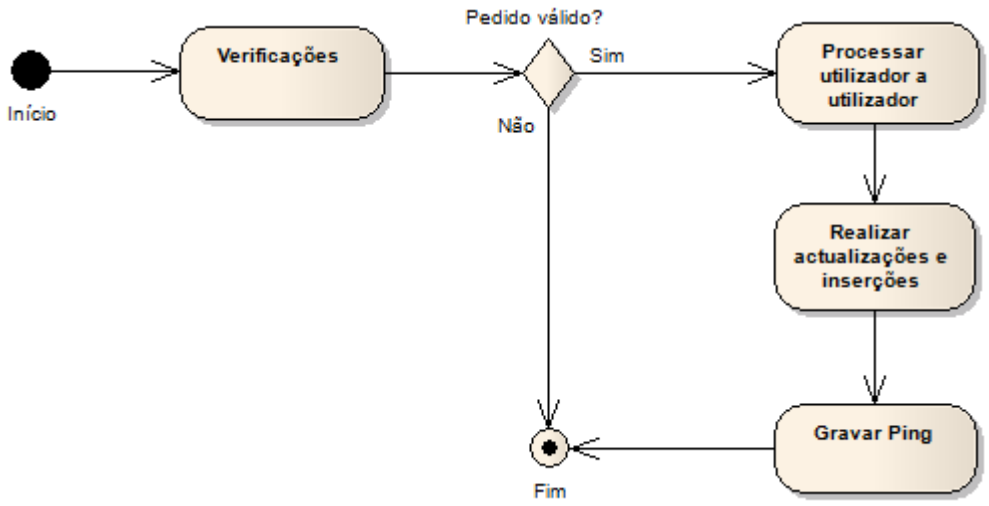


Figura 3.5: Acções do serviço *put* lookup.

necessidade que o serviço tenha uma *performance* elevada. Abordo este tema em maior profundidade na secção 3.3.5.

Para cada utilizador é efectuado um conjunto de acções. No caso de ser um possível utilizador, actualiza as informações destes na BD. Caso seja um utilizador e se encontre assinalado na BD para ser verificado, poderão ser actualizadas informações deste, inseridas novas estatísticas e *tweets* e modificada a prioridade. Este processo encontra-se ilustrado na figura 3.6.

3.3.2 Serviços links

Os serviços descritos na presente secção são responsáveis pela interacção entre clientes links e o servidor.

Get links

Este serviço gera uma lista de utilizadores de quem o servidor pretende as relações. Esta lista é gerada com o mesmo algoritmo de prioridades usado no serviço *get* lookup, usando dados de entrada diferentes.

Put links

Este serviço recebe dados enviados pelos clientes links. Dados esses referentes às relações de utilizadores. Possui uma arquitectura similar ao serviço *put* lookup (ver figura 3.5), sendo que, as acções efectuadas para cada utilizador são diferentes. Para cada utilizador, insere a lista de seguidores caso esta seja diferente da última lista de seguidores presente na BD para o utilizador em questão. Age de forma similar para a lista de amigos. O processo encontra-se ilustrado na figura 3.7.

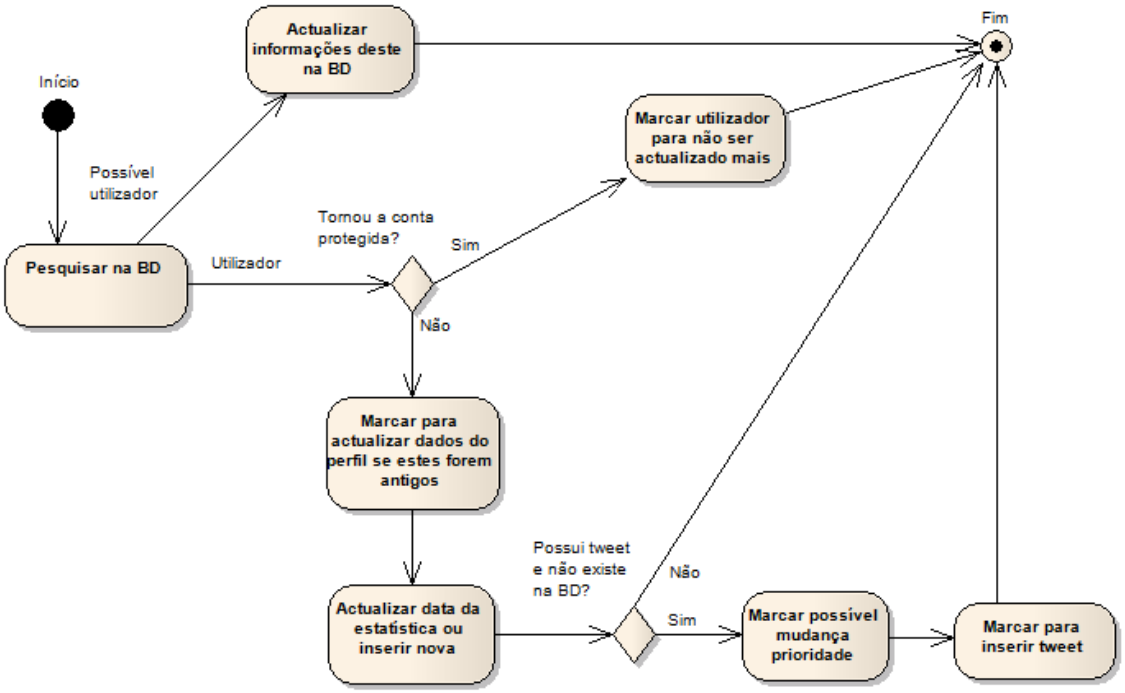


Figura 3.6: Acções efectuadas pelo serviço *put* lookup para cada utilizador.

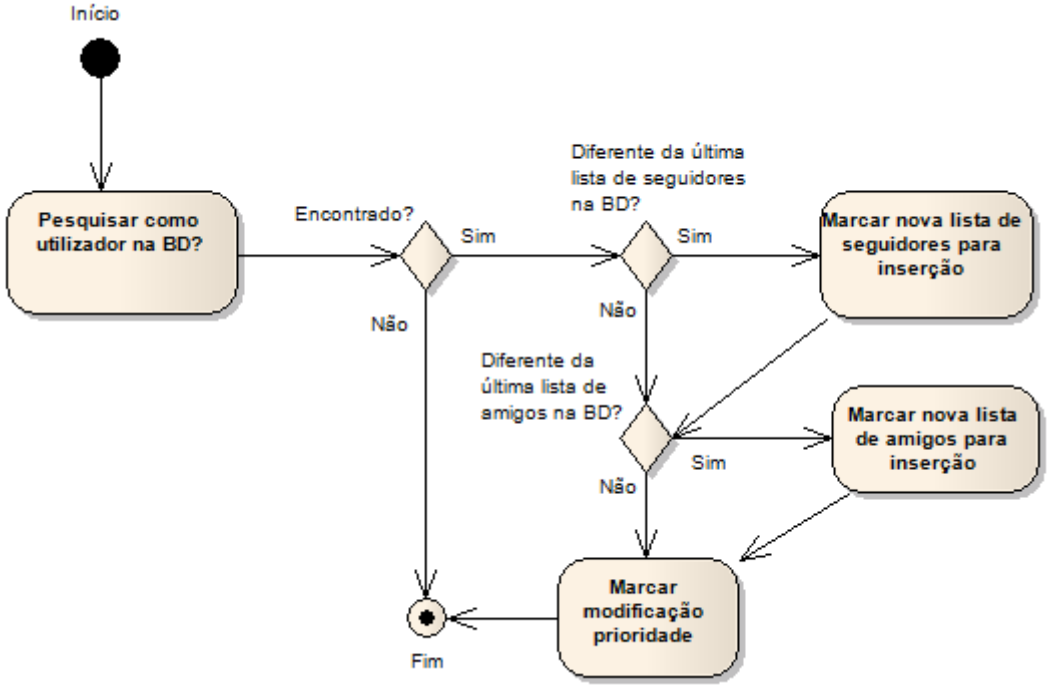


Figura 3.7: Acções efectuadas pelo serviço *put* links para cada utilizador.

3.3.3 Serviço erros

O servidor possui um serviço *put* para os clientes reportarem erros, especificando um texto e um código numérico.

3.3.4 Escalonamento

O servidor deve coordenar os clientes, de forma a estes verificarem mais vezes alguns utilizadores que outros. Torna-se, portanto, necessário dividir os utilizadores em classes distintas.

Cada utilizador possui associado a si prioridades. Os serviços *get*, geram a lista de utilizadores dividindo esta em várias sub-listas, dando ênfase aos utilizadores com maior prioridade. Dentro de uma classe de prioridades os utilizadores que não recebem dados há mais tempo são seleccionados primeiro. Isto resulta em que utilizadores com prioridades elevadas são verificados mais vezes que utilizadores com prioridades inferiores. Na figura 3.8 está disponível um exemplo simplificado dos utilizadores que o processo de escalonamento poderia entregar a um cliente.

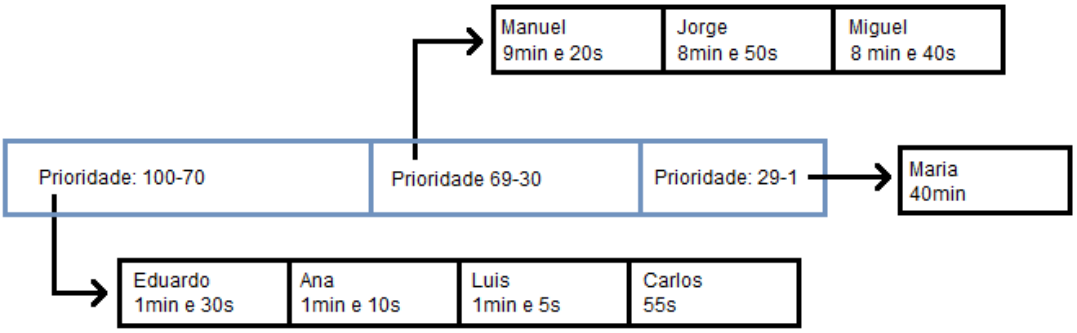


Figura 3.8: Exemplo simplificado do processo de escalonamento. Os tempos presentes na figura referem-se à última vez que os utilizadores foram verificados.

Os utilizadores enviados para um cliente ficam alocados por um período de tempo, não podendo ser enviados a outros clientes nesse período de tempo. Isto permite uma maior rotatividade e optimização de recursos (evita que vários clientes peçam informações sobre o mesmo utilizador ao mesmo tempo).

A prioridade é calculada de forma diferente nos serviços de lookup e links.

Lookup

No caso do cliente lookup que recolhe *tweets* é óbvio que os utilizadores mais activos (publicam mais) têm que ser verificados mais vezes. Cada *tweet* recebido é pesquisado

na BD. Caso não seja encontrado a prioridade do utilizador é aumentada. Caso contrário diminuída. Desta forma privilegia-se os utilizadores que publicam mais.

O sistema foi melhorado progressivamente tendo como base dados reais recolhidos pelo TwitterEcho. Durante o período nocturno existe pouca actividade no Twitter, como pode ser observado na figura 3.9.

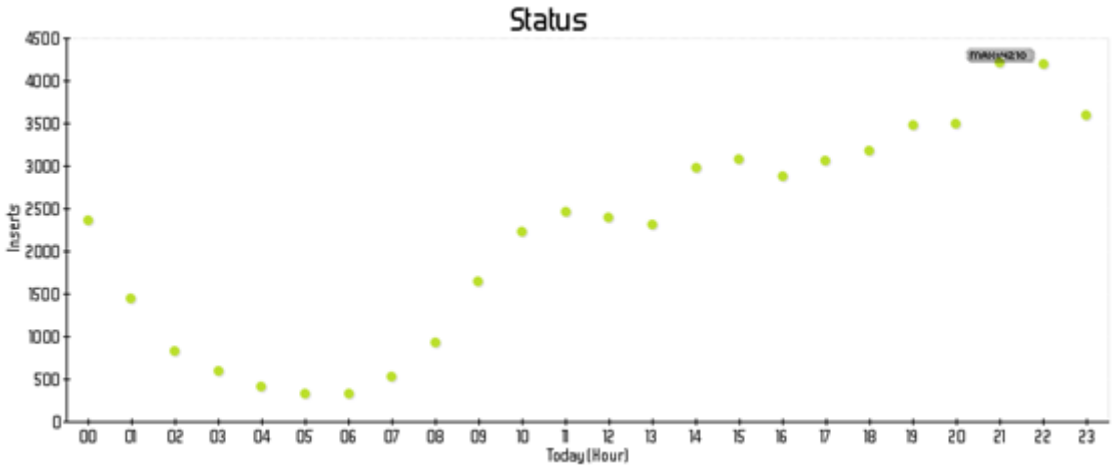


Figura 3.9: Actividade diária (número de *tweets* por hora).

A actividade no Twitter varia ao longo do dia, portanto não faz tanto sentido diminuir prioridades no período nocturno cuja actividade é diminuta. Adicionei também a possibilidade de especificar um tempo mínimo que deve distar a data do último *tweet* recolhido da data actual para ocorrer uma diminuição de prioridade. O Twitter possui também um conjunto grande de utilizadores inactivos (não publicam). Estes utilizadores inscrevem-se e utilizam o Twitter, por algum tempo, deixando de o utilizar sem apagar conta ou utilizadores que apenas usam o Twitter para seguir outros utilizadores e se manterem actualizados. Estes utilizadores são marcados como inactivos permitindo a optimização de recursos (abordo este tema em maior detalhe na secção 3.4.7).

O processo de escalonamento é de extrema importância. Na presente data na BD do TwitterEcho, 19841 utilizadores publicaram pelo menos duas vezes, num total de 2,206,519 *tweets*. Destes utilizadores apenas 451 publicaram pelo menos mil vezes, no total 826,925 *tweets*. Isto significa que aproximadamente 2,2% dos utilizadores publicaram cerca de 37% dos *tweets*. Os utilizadores mais activos têm que ser verificados com maior frequência de forma a evitar perdas de *tweets*.

No lookup existem 6 classes de prioridades: 100, 99-81, 80-61, 60-41, 40-21 e 20-1. Isto permite dividir os utilizadores do Twitter em várias classes de actividade, por exemplo alguém com prioridade 100 publica bastante e muito recentemente, já alguém com prioridade 60 publicou há algum tempo mas não tão recentemente.

Links

Quando o serviço *put* links recebe os seguidores e amigos, para cada utilizador compara com a última ocorrência na BD, caso seja diferente aumenta a prioridade do utilizador, caso contrário diminui. No links existem 3 classes de prioridades 100-61, 60-21 e 20-1.

3.3.5 *Performance*

Os serviços referidos no presente capítulo encontram-se em uso por vezes por dezenas de clientes. Os serviços de lookup são acedidos cinco vezes por minuto e os de links uma vez a cada dez minuto por cada cliente. Devido a esse facto, a velocidade de execução é de grande importância. Os tempos de execução dos serviços são gravados de forma a possibilitar monitorizar e avaliar o impacto de mudanças no código na *performance*.

Os tempos de execução dos *gets*, tal como esperado, são similares. O *get* lookup é um pouco mais lento porque possui um maior número de classes de prioridades, logo faz mais pedidos ao MySQL. O *put* lookup executa centenas de pesquisas na BD. Como forma de aumentar a *performance* deste serviço as inserções e actualizações são agrupadas de forma a minimizar o número de chamadas ao MySQL. O tempo de execução do *put* links possui uma grande dependência da quantidade de dados recebidos, devido à inserção na BD (existem utilizadores do Twitter com um número enorme de seguidores).

Os tempos de execução dos serviços, regra geral, são bastante satisfatórios. No entanto ocorrem por vezes tempos de execução não aceitáveis, isto deve-se ao facto de a *performance* se encontrar dependente das capacidades do servidor, sendo que este é partilhado (utilizado por outros sistemas).

3.3.6 Trabalho futuro

As grandes melhorias que podem ser desenvolvidas nos serviços dizem respeito a um aumento da *performance* destes. Estas melhorias são ainda mais importantes caso se pretenda modificar o foco deste sistema de forma a recolher dados de uma comunidade de dimensão superior à actual. Deve ser dada especial atenção aos serviços de lookup que possuem uma utilização mais frequente. Existem três principais formas de diminuir a carga de trabalho dos serviços: serem executados menos vezes, efectuar menos operações, e operações mais rápidas.

Cada serviço de lookup é acedido por cada cliente cinco vezes por minuto. É realizado um *get* para obter os utilizadores de quem se pretende informações e enviadas as informações destes para o serviço *put*, ou seja, cada cliente lookup executa cinco vezes por minuto as interacções presentes na figura 3.3. A arquitectura pode ser alterada para

apenas executar um *get* por um maior número de utilizadores e depois os *put's* correspondentes. Desta forma consegue-se uma diminuição significativa do número de vezes que o serviço *get* é executado; com a desvantagem que desta forma ficam alocados a um cliente mais utilizadores por um maior período de tempo. Pode portanto ocorrer diminuição de eficiência na recolha de *tweets*. No entanto, não considero que esta diminuição seja significativa.

O serviço que executa maior número de operações na BD é o *put* lookup. Para cada utilizador e *tweet* enviado executa o conjunto de operações presentes na figura 3.6. Uma das formas de diminuir o número de operações é os clientes realizarem algum pré-processamento. Por exemplo se o servidor manter uma lista actualizada contendo para cada utilizador o identificador do último *tweet*, os clientes podem consultar esta lista e não enviar *tweets* que já existem na BD. Desta forma trocar-se-ia centenas de pesquisas na BD por apenas uma a cada minuto, de forma a manter a lista actualizada.

Uma das formas mais usuais de tornar as operações mais rápidas é garantir que todas as pesquisas efectuadas na BD usam índices para o efeito. Isso acontece nos serviços de *put* mas não nos de *get*. Efectuei testes com o objectivo de verificar se tornar o campo prioridade índice da tabela teria impacto significativo no tempo de execução. O impacto foi praticamente inexistente. Isto deve-se ao facto que a tabela utilizadores é de pequena dimensão (menos de cem mil entradas). No entanto, numa comunidade de maior dimensão provavelmente teria efeito.

3.4 Módulos

Os módulos são responsáveis por diversas tarefas. Estes encontram-se agrupados em áreas de competência na seguinte figura:

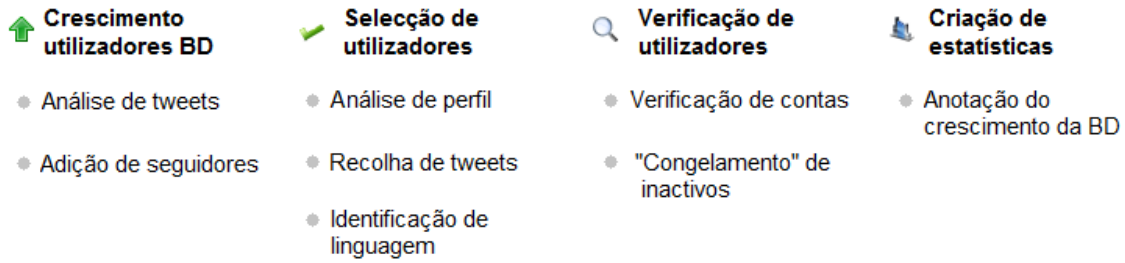


Figura 3.10: Módulos agrupados por área de competência.

O grupo de crescimento de utilizadores diz respeito aos módulos responsáveis pelo crescimento dos utilizadores presentes na BD. O módulo de análise de *tweets* extrai utilizadores mencionados e adiciona-os como possíveis utilizadores a obter informação. O

módulo de adição de seguidores adiciona seguidores de utilizadores de quem o sistema actualmente recolhe dados.

O grupo de selecção de utilizadores selecciona os utilizadores portugueses dos utilizadores obtidos pelo grupo anterior. O módulo análise do perfil realiza uma análise do fuso horário, localização e nome. O módulo identificação de linguagem procura identificar se a língua é portuguesa a partir de um conjunto de *tweets* recolhidos pelo módulo recolha de *tweets*.

O grupo de verificação de utilizadores é responsável por efectuar verificações na lista de utilizadores de quem se recolhe informações. O módulo verificação de contas verifica se as contas continuam válidas para obter dados, não foram suspensas ou apagadas. O módulo "congelamento" de inactivos coloca utilizadores em *stand-by* para serem verificados em períodos nocturnos de pouca actividade.

O grupo de criação de estatísticas possui o módulo anotação do crescimento da BD, tem como objectivo registar a evolução do número de registos das tabelas da BD ao longo do tempo.

Os módulos acima descritos são executados no servidor de forma periódica usando o *crontab* [Cho]. Nas próximas secções abordarei em maior detalhe o objectivo e funcionamento destes módulos.

3.4.1 Análise de *tweets*

É fundamental o crescimento do número de utilizadores de quem se pretende informações. A situação ideal a atingir seria possuir a lista de todos os portugueses que possuem Twitter.

Quando um utilizador português menciona alguém ou responde a algum utilizador pressupõe-se uma probabilidade elevada de este ser Português. Para confirmar esta suposição, seleccionei de forma aleatória cinquenta utilizadores da BD. Para cada utilizador verifiquei manualmente que realmente são portugueses. Dos cinquenta utilizadores quarenta e sete são realmente portugueses. Destes quarenta e sete seleccionei de forma aleatória cem *tweets* que contivessem menções. Dos *tweets* extrai os nomes de utilizadores mencionados. Manualmente identifiquei a nacionalidade, sendo que 79% destes são portugueses.

Devido ao que anteriormente referi, a análise de *tweets* com o objectivo de extrair e adicionar utilizadores mencionados é uma óptima forma de fazer crescer a BD com qualidade. No entanto, este método possui uma margem de erro razoável, sendo que por si só não é suficiente.

O módulo de análise de *tweets* só adiciona utilizadores com conta válida, neste contexto uma conta é considerada válida quando é possível obter *tweets* desta, ou seja, não pode ser estar protegida nem suspensa.

A primeira implementação deste módulo para cada utilizador extraído pesquisa se este se encontra na BD como utilizador. Caso não seja encontrado pesquisa como possível utilizador. Se o utilizador for encontrado na BD (como utilizador ou possível utilizador) modifica o número de menções deste. Caso contrário, acede a página deste no Twitter e se for uma conta válida adiciona-o.

Produz os resultados esperados mas efectua demasiadas operações na BD, analisando um exemplo de 8 *tweets*:

1. Boa tarde ;) @**jallima** ...
2. @**jallima** olá migo! Bons olhos te leiam migo! Tudo bom? :)
3. @**jallima** Claro, tem de ser!
4. pronto, regressada do Porto! =)
5. @**jallima** :)
6. @**Rykardow** @**saronaa** Bom, mas agora vou sair, see you next week twittrers xD
7. também quero... RT @**saronaa**: @**ay_cee** mandei-te email
8. @**Rykardow** tagarela (@**PippaFerreira**)

Supondo que nenhum dos utilizadores acima mencionados se encontra na BD, a execução vai originar para cada *tweet* as seguintes acções na BD:

1. Duas pesquisas, uma inserção (como possível utilizador) e uma actualização;
2. Duas pesquisas e duas actualizações;
3. Duas pesquisas e duas actualizações;
4. Sem acções;
5. Duas pesquisas e duas actualizações;
6. Quatro pesquisas, duas inserções e uma actualização;
7. Quatro pesquisas, uma inserção e duas actualizações;
8. Quatro pesquisas, uma inserção e duas actualizações.

De forma a melhorar a *performance* implementei um novo algoritmo. Primeiro analisa todos os *tweets* e guarda os resultados numa *hash table*. No fim executa as pesquisas, actualizações e inserções na BD. As actualizações e inserções são efectuadas num pedido cada. Desta forma o número de pedidos de actualização e inserção não depende

do número de *tweets* analisados. O número de pesquisas apenas depende do número de utilizadores mencionados distintos, ao invés do número de utilizadores mencionados. No exemplo acima enunciado esta implementação executa dez pesquisas, uma inserção e duas actualizações. No total treze chamadas contra as trinta e sete da solução anterior.

O sistema recolhe milhares de *tweets* por dia e todos têm que ser analisados. Obter dados de cada utilizador requer aceder à página do Twitter. Este processo é demorado e encontra-se dependente da velocidade de resposta do Twitter. Isto torna a *performance* deste processo importante, o que me levou a optar por uma arquitectura um pouco mais complexa mas mais rápida.

Devido ao que referi anteriormente espera-se que o tempo de execução deste processo esteja intimamente relacionado com o tempo de resposta do Twitter e número de utilizadores de quem é necessário obter dados. Essa relação é visível na figura 3.11.

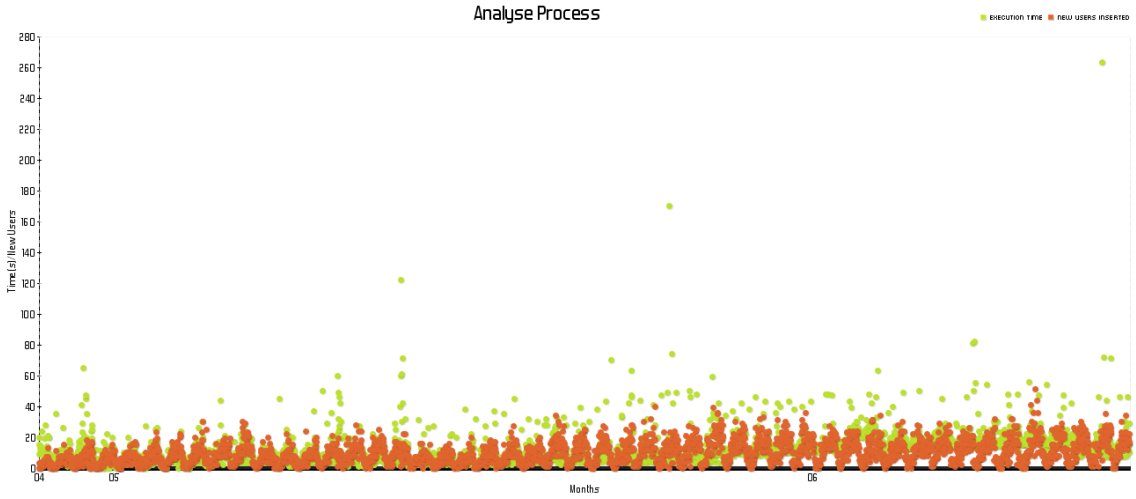


Figura 3.11: Relação entre o tempo de execução de processo de análise de *tweets* e o número de utilizadores inseridos (o tempo de execução encontra-se com cor clara e o número de utilizadores inseridos cor escura). A sobreposição é visível.

Actualmente este processo consegue analisar até cerca de 120 *tweets* por minuto, o que totaliza aproximadamente 170 mil *tweets* por dia, bem acima das necessidades actuais, ilustradas na figura 3.12.

No caso de se pretender aplicar de futuro este sistema numa comunidade de maior dimensão, cuja recolha seja superior a 170 mil *tweets* por dia, é necessário reduzir o número de acessos a páginas do Twitter. Isto pode ser conseguido mantendo na BD uma lista de contas inválidas, sendo esta refrescada e actualizada periodicamente.

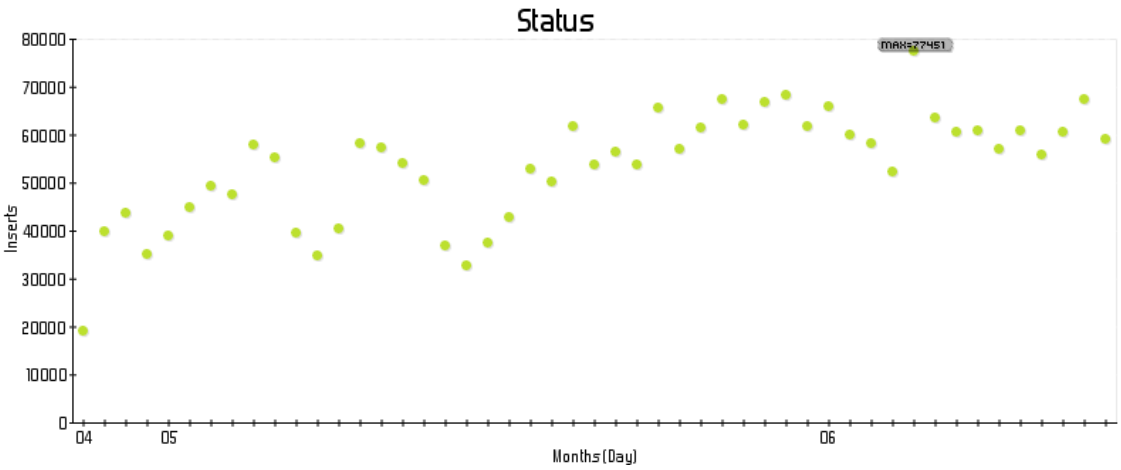


Figura 3.12: Número de *tweets* recolhidos por dia.

3.4.2 Adição de seguidores

O módulo análise de *tweets* possui uma limitação evidente: todos os utilizadores que não sejam mencionados por utilizadores presentes na BD não são adicionados. Surge portanto a necessidade de adicionar utilizadores de outras formas. Este módulo tem como objectivo adicionar seguidores de utilizadores da BD como possíveis utilizadores.

De forma abreviada este serviço: selecciona de forma aleatória utilizadores da BD, e obtém os seguidores destes recorrendo à API do Twitter. O módulo utiliza o método *Statuses/followers* (secção A.1.6). Para cada seguidor efectua um conjunto de verificações e caso este passe nas verificações, adiciona-o como possível utilizador. A sequência das acções pode ser observada na figura 3.13.

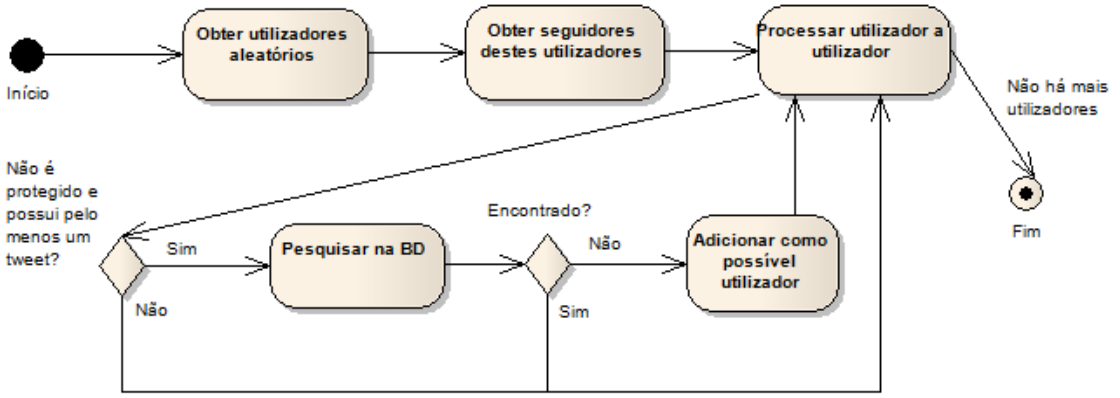


Figura 3.13: Implementação do módulo adicionar seguidores.

Este módulo usa a API do Twitter estando sujeito aos limites de uso impostos pelo Twitter. De forma a não se encontrar limitado pela API do Twitter deveria pedir dados

aos clientes. No entanto numa comunidade pequena como a portuguesa não existe uma necessidade forte de implementar esta solução mais complexa. É uma melhoria fundamental caso o foco mude para uma comunidade de dimensão superior.

3.4.3 Análise de perfil

Os módulos referidos anteriormente permitem recolher novos utilizadores, mas é necessário identificar os portugueses. O presente módulo tem como objectivo efectuar uma selecção destes de acordo com a informação presente no perfil. Os campos de perfil avaliados são: fuso horário, localização e nome.

Possui duas funções distintas: tentar provar que o utilizador é português e tentar provar que possui outra nacionalidade.

A prova de que um utilizador é português processa-se da seguinte forma:

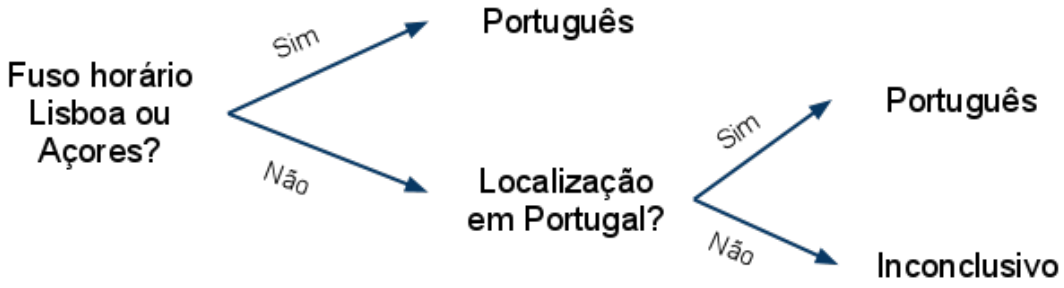


Figura 3.14: Algoritmo decisão se um utilizador é português.

A prova que um utilizador não é português processa-se da seguinte forma:

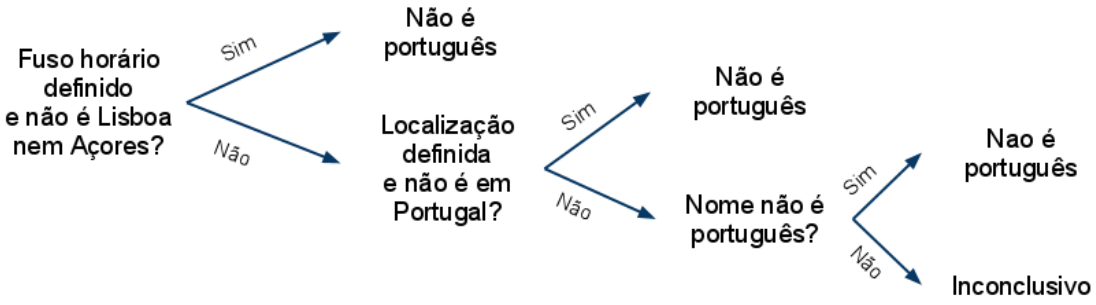


Figura 3.15: Algoritmo decisão se um utilizador não é português.

Estes dois métodos são combinados para gerar um resultado (ver tabela 3.1).

É português (fig. 3.14)?	**Não é português (fig. 3.15)?**	**Resultado**
Inconclusivo	Inconclusivo	Inconclusivo
Inconclusivo	Sim	Não é português
Sim	Inconclusivo	Português
Sim	Sim	Inconclusivo

Tabela 3.1: Combinação dos métodos 3.14 e 3.15.

Na BD estão guardadas as seguintes listas: cidades e locais em Portugal, cidades e locais em outros países e nomes não portugueses. Isto permite que sejam facilmente adicionadas novas cidades e nomes de forma a melhorar o módulo de análise de perfil. Permite também facilmente adaptar este módulo para outro país mudando apenas os dados na BD sem ser necessário modificar o algoritmo.

A lista de localidades de Portugal com título de cidade encontra-se disponível na *Web* e foi a usada na análise de perfil. A lista de cidades de outros países foi criada com base em listas de cidades com forte presença no Twitter disponíveis na *Web* [Gra]. A lista de nomes não portugueses foi criada tendo como base uma de lista de nomes disponibilizada pelo Portal do Cidadão [dC]. A lista contém nomes aceites e não aceites para registo de recém-nascidos.

Este módulo consegue classificar uma percentagem significativa de utilizadores (80% classificados e 20% inconclusivos), de futuro aproveitando os dados disponíveis de mais de meio milhão de possíveis utilizadores podem ser retiradas localidades não presentes nas listas e inseridas na BD. Este trabalho permitirá a classificação de mais utilizadores apenas com base na informação disponível no perfil no Twitter.

3.4.4 Recolha de *tweets*

O módulo de análise de perfil não permite classificar todos os utilizadores, os utilizadores que não foram classificados pelo perfil podem ser classificados tendo em conta o conteúdo dos *tweets*. Este módulo tem como objectivo recolher *tweets* dos utilizadores cuja análise de perfil foi inconclusiva.

É escolhido um conjunto de utilizadores de forma aleatória e são pedidos *tweets* destes à Search API, caso estes *tweets* ainda não se encontrem na BD são inseridos.

Este módulo efectua cerca de cem mil pedidos por dia ao Twitter, é suficiente para obter *tweets* de um grande número de utilizadores num curto espaço de tempo. Mas devido a fazer uso da Search API são perdidos *tweets* (secção A.2). Para evitar a perda de *tweets* pode ser usada a REST API (método *Statuses/user_timeline* A.1.2) que não filtra por relevância e não devolve apenas *tweets* recentes, devolvendo até 200 *tweets*, mas esta possui um limite de chamadas baixo (secção A.4).

Uma melhoria seria tornar este módulo num serviço. Os clientes passariam a ser responsáveis por recolher *tweets* dos possíveis utilizadores. Esta mudança requereria o aumento do número de clientes e consequentemente o número de máquinas. Pode ser implementada uma solução híbrida, o servidor usaria a Search API e apenas caso os *tweets* devolvidos não sejam suficientes para análise, os utilizadores passariam a ser enviados a clientes.

3.4.5 Identificação de linguagem

O módulo de análise de perfil não consegue classificar todos os utilizadores, é necessário classificar estes de outra forma. Escolhi analisar os *tweets* de forma a identificar a linguagem. O objectivo deste módulo é dado um conjunto de *tweets* identificar se estes se encontram escritos em português de Portugal.

Os *tweets* possuem diversas características (os valores foram calculados tendo como base 2,5 milhões de *tweets* recolhidos pelo TwitterEcho):

- Os *tweets* são curtos, em média 81 caracteres.
- Uma percentagem significativa de *tweets* possui URL's, aproximadamente 40%.
- Uma percentagem significativa de *tweets* possui menções, aproximadamente 48%.
- Alguns *tweets* possuem marcadores/*hashtags* (secção 1.2.2), aproximadamente 14%.
- O Twitter é usado em diversos países, logo existem *tweets* em várias línguas (asiáticas também).
- Os *tweets* possuem erros ortográficos.
- Os *tweets* em português nem sempre possuem acentuação.
- Após remoção de ULR's, menções, marcadores, espaços, entre outros, os *tweets* ficam ainda mais curtos, com uma média de 58 caracteres.

Tendo em conta que o módulo de recolha de *tweets* na melhor das hipóteses recolhe cem *tweets* para um utilizador existe pouco texto para efectuar a identificação.

A identificação de linguagem foi abordada ao longo do tempo de diversas formas. Na revisão bibliográfica referi:

- Combinações únicas de letras (secção 2.2.1);
- Palavras comuns curtas (secção 2.2.2);
- N-Grams (secção 2.2.3);

- Abordagens baseadas em compressão (secção 2.2.4);
- Abordagem estatística (secção 2.2.5).

A combinação única de letras baseia-se num conjunto de combinações únicas de cada língua. Em textos curtos, como *tweets*, as combinações podem não aparecer e caso apareçam podem não ser suficientes para identificar uma língua. Por exemplo a cidade Montreux possui a combinação única *eux* do Francês, no entanto como é uma cidade pode ser usada em várias línguas.

As palavras comuns curtas são usadas com sucesso na classificação de textos com alguma dimensão. Os erros ortográficos afectam bastante este método. Obriga também à divisão em palavras, o que é difícil em algumas línguas asiáticas. Neste contexto os textos são curtos, existem erros ortográficos e *tweets* em línguas asiáticas.

A identificação de linguagem por N-Grams (segmentos de N caracteres) possui bons resultados em textos curtos. Erros ortográficos apenas afectam uma parte dos N-Grams deixando os outros intactos. Obriga à divisão em palavras.

Abordagens baseadas em compressão possuem bons resultados em textos curtos. Compressão é uma operação pesada. Não possui tantos estudos e trabalhos desenvolvidos como N-grams e não está demonstrado que possuam resultados superiores. Teahan e Harper [TH] usaram documentos de treino até 1,5MB. É um tamanho considerável. A vantagem é que não necessita de divisão em palavras.

Abordagem estatística não necessita de pré-processamento para dividir o texto em palavras, o próximo estado depende apenas do actual. Dunning [Dun94] conseguiu bons resultados com documentos de treino inferiores a 50000 bytes (aproximadamente 390KB).

Optei por N-grams devido a adequarem-se ao contexto actual e possuírem vários estudos e trabalhos [CT94] [Pra99] [ACT04] [Elw99].

Foquei-me nos uni gramas dessa forma evito a divisão em palavras. Por exemplo a distribuição das letras no francês encontra-se representada na figura 3.16.

Na figura 3.17 encontram-se representadas as frequências das letras de cem *tweets* em francês de um utilizador (os espaços são ignorados na contagem).

As semelhanças entre a figura 3.16 e 3.17 são óbvias. No entanto, é possível verificar que por exemplo as letras K e W possuem uma frequência elevada. Isso deve-se à sua ocorrência em URL's, nomes de utilizadores, tópicos, entre outros. Desenvolvi um algoritmo para pré-processamento dos *tweets*. Remove URL's, menções, *hashtags*, transforma as letras acentuadas em letras similares (ex: á em a), transforma maiúsculas em minúsculas e retira todos os caracteres que não sejam letras e espaços.

Na figura 3.18 encontra-se ilustrada a frequência das letras após aplicação do algoritmo de pré-processamento. É possível verificar que as semelhanças se mantêm e o número de ocorrências das letras K e W diminui.

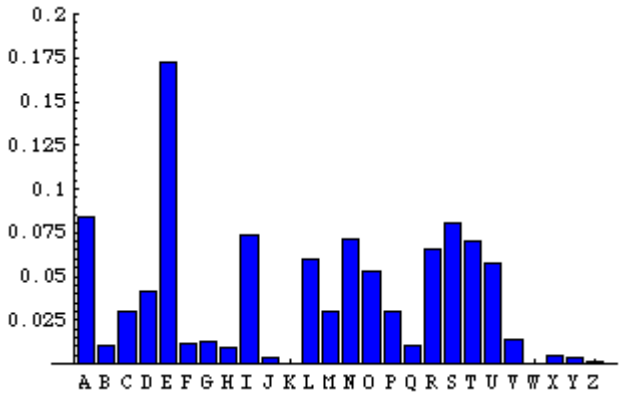


Figura 3.16: Frequência das letras no francês calculadas através de vários textos no total cem mil letras foram tidas em conta [el].

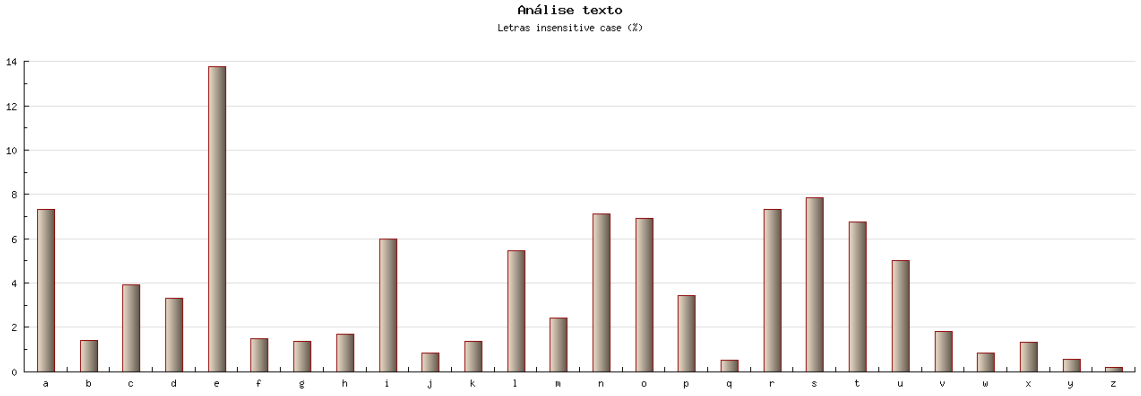


Figura 3.17: Frequência das letras de cem *tweets* em francês de um utilizador.

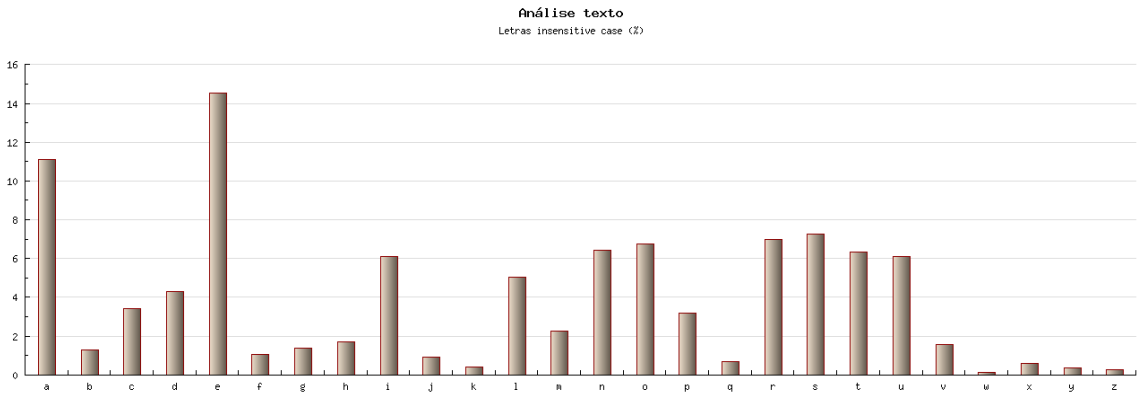


Figura 3.18: Frequência das letras de cem *tweets* em francês de um utilizador.

Escolhi o francês porque tal como o português deriva do latim, por isso estas duas línguas aproximam-se. No entanto, existem diferenças entre a frequência de letras do

francês (figura 3.16) e a frequência de letras no português (figura 3.19).

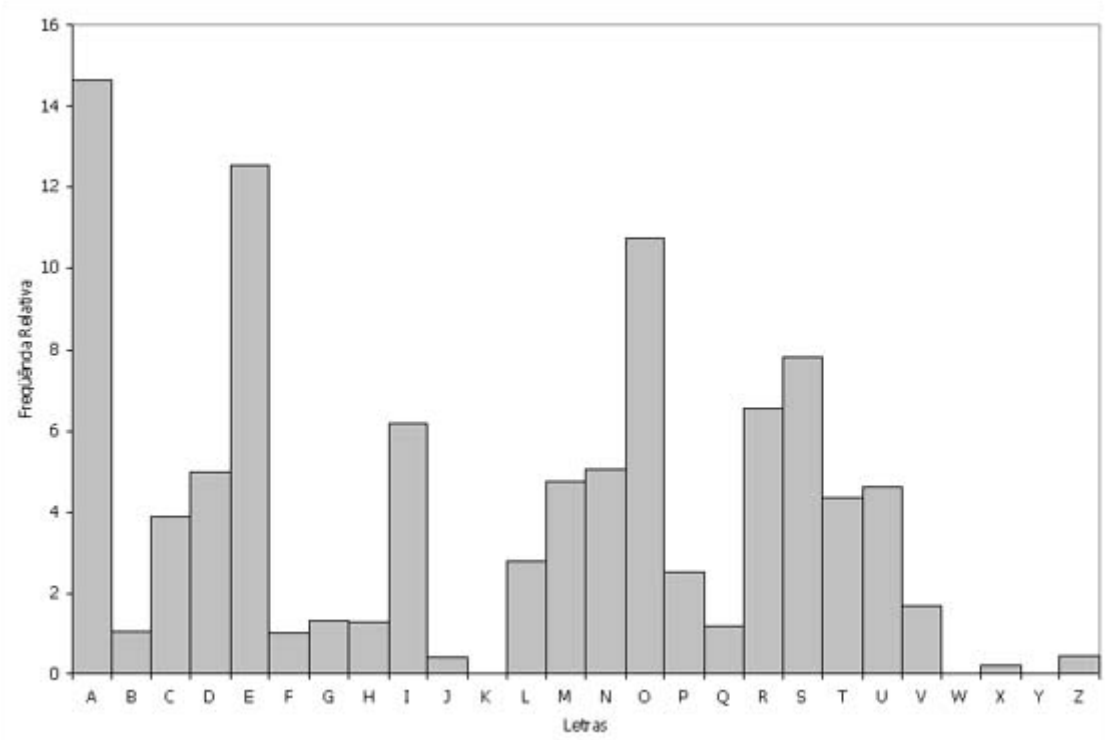


Figura 3.19: Frequência das letras no português. Foram usados seis autores conhecidos de épocas diferentes, no total 725511 letras. Apenas um autor é português os restantes são brasileiros. Os textos em português do Brasil representam 98% do texto total [el].

As diferenças nas frequências de letras do português para o francês são significativas. Escolhi de forma aleatória um utilizador que publica em português de Portugal, usei trinta *tweets* (755 bytes). As frequências das letras após pré-processamento encontram-se representadas na figura 3.20.

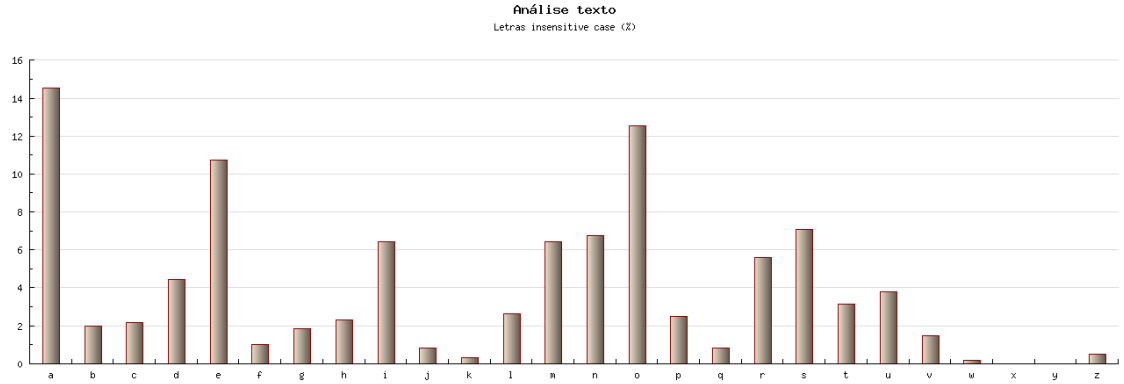


Figura 3.20: Análise de frequência de trinta *tweets* em português de Portugal..

É possível concluir que os *tweets* se encontram escritos em português devido às semelhanças entre a figura 3.20 e a figura 3.19.

Os *tweets* estão escritos em português de Portugal e a frequência de letras presentes na figura 3.19 é maioritariamente em português do Brasil. As frequências são similares, isto cria um problema, a população do Brasil é aproximadamente dezanove vezes superior à população de Portugal. O Brasil possui por isso uma representação grande no Twitter. Torna-se necessário distinguir entre português de Portugal e do Brasil. O uso de uni gramas não é suficiente, isto indicia que o uso de N-grams para distinção entre português de Portugal e do Brasil pode não ser indicado. Cavnar e Trenkle [CT94] obtiveram os piores resultados na identificação de línguas, no caso da identificação de português do Brasil. Como forma de verificar estudei os digramas mais frequentes para conjuntos grandes de *tweets* para um utilizador que escreve em português de Portugal (figura 3.21) e outro que escreve em português do Brasil (figura 3.22).

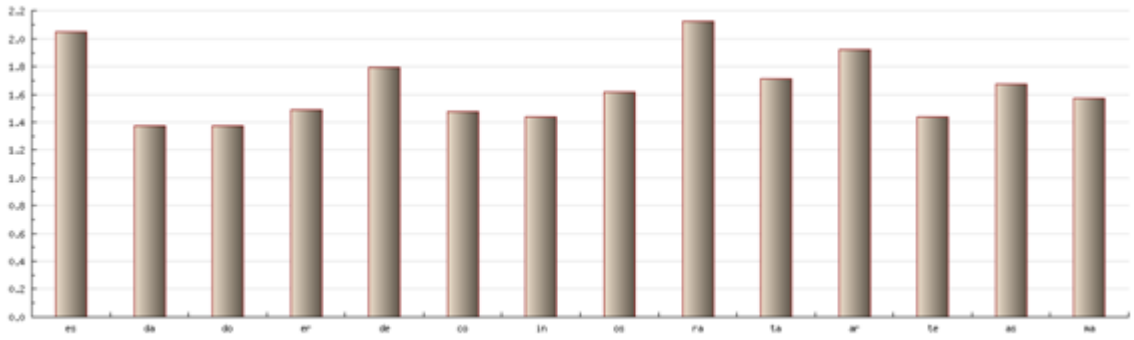


Figura 3.21: Digramas de português de Portugal (**es**, **da**, do, er, **de**, co, in, os, **ra**, **ta**, **ar**, **te**, as, **ma**). Conjunto de *tweets* de tamanho 274KB.

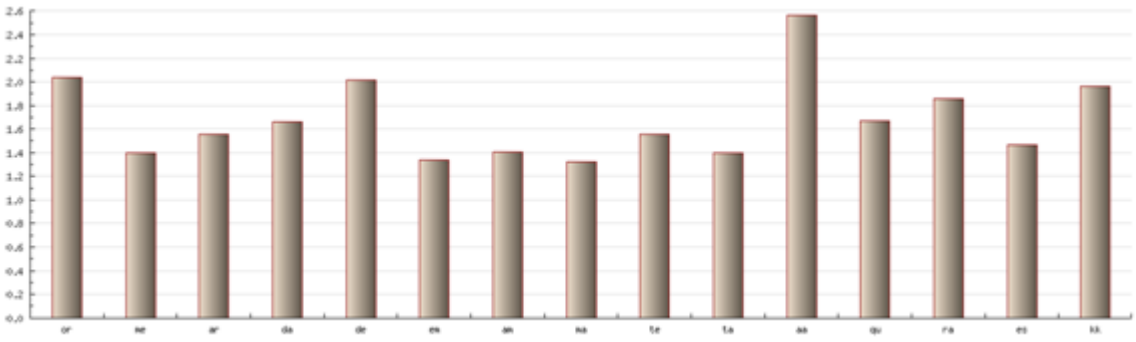


Figura 3.22: Digramas de português do Brasil (or, me, **ar**, da, **de**, *em*, am, **ma**, **te**, **ta**, aa, qu, **ra**, **es**, kk). Conjunto de *tweets* de tamanho 63KB.

Nos digramas mais frequentes na amostra de português de Portugal 8 em 14 aparecem na amostra de português do Brasil, é um número significativo. No entanto, o português de

Portugal e do Brasil é distinguível de outras formas. Os seguintes *tweets* foram publicados pelo utilizador cujos digramas estão presentes na figura 3.22:

1. @KATEVOADOIS aqui diz: tenta**ndo** conexão com o **usuario kkkkkkk**
2. @virginialara eu ja perguntei se ela ta chateada comigo quetem um mês que não me fala, ela nem responde...vou morrer =(**rsrsrsr**
3. @CarolLemosRJ **poxa sério**???porquê ela fez isso? tudo bem cmg, ela nem me conhece, mas tudo q fazes tanto por ela..não entendo isso...
4. @anaclaudiam35 oi amor tudo bem e contigo?
5. @ISAmrVerdadeiro own que lindo tou doida para ver,quero ir para casa agoraaaaa #parabensivete ela merece tuuudooooo

No primeiro *tweet* é usado o gerúndio (tentando), uma palavra que não existe em português de Portugal (*usuário*) e uma forma de rir (*kkkkkkk*) diferente do tradicional *LOL* usado em Portugal. No segundo *tweet* aparece *rsrsrsr* outra forma de "rir" típica dos brasileiros. No terceiro *tweet* uma expressão usual no Brasil ("*poxa* sério?"). Os últimos dois *tweets* estão relacionados com a emotividade e descontracção, própria dos brasileiros, que se reflecte na escrita, prolongam as palavras (agoraaaaa e tuuudooooo) e tem um à vontade enorme ("oi amor tudo bem e contigo?").

Agregando todo o conhecimento transmitido até agora de identificação de linguagem, e tendo como base o trabalho desenvolvido por Cavnar e Trenkle [CT94] primeiro criei o perfil português. Como base utilizei quinze mil *tweets*, executei pesquisas por palavras comuns e trigramas comuns em Inglês [We] [Cow]. No final usei aproximadamente dez mil *tweets*. Apliquei o algoritmo de pré processamento para retirar URL's, menções, entre outros, resultando num perfil de aproximadamente 50KB (Cavnar e Trenkle [CT94] usaram perfis entre 20-120KB). Para criar o perfil optei por usar *tweets* e não livros ou notícias, a forma de escrever é distinta e poderia resultar em diferenças significativas. Foi efectuada uma contagem de ocorrência das letras, sendo esta gravada num ficheiro por ordem de maior ocorrência para menor ocorrência de todas as letras. Este ficheiro é o perfil da língua portuguesa usado por este módulo em todas as identificações.

Quando é necessário classificar uma amostra é utilizado o mesmo processo usado na criação do perfil. Depois é efectuado o cálculo da distância (exemplo figura 3.23).

O cálculo final é dado pela soma do valor de fora de ordem para todas as letras. Este método de cálculo de distância é baseado no usado por Cavnar e Trenkle [CT94]. Existe a possibilidade se uma amostra for pequena de uma letra não existir, Cavnar e Trenkle [CT94] neste caso atribuem um peso máximo, igual em qualquer situação. Optei por atribuir um peso em função da importância da letra. Por exemplo a letra o é de uso comum no português, ou seja, esta não se encontrar presente indicia que o texto não se

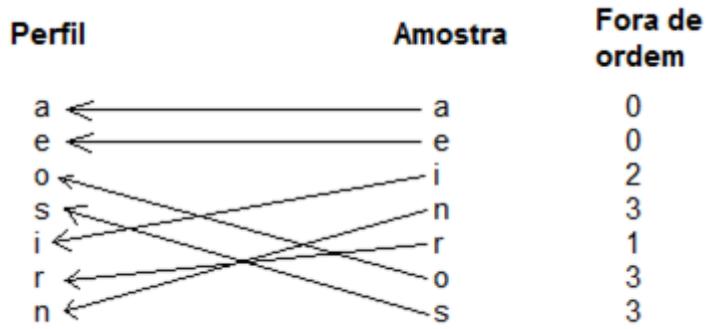


Figura 3.23: Exemplo de parte do cálculo da distância. O perfil é apenas ilustrativo e não corresponde ao em uso.

encontra escrito em português. Por sua vez a não ocorrência da letra z não é significativo. Em vez de um máximo atribuo um valor diferente para cada letra em falta, este valor é calculado através do perfil. A letra mais frequente no perfil vale 26, a segunda 25, a terceira 24 e por aí adiante.

No caso do cálculo final ser inferior a um determinado valor (que foi determinado com base em testes manuais), a língua é considerada portuguesa.

A segunda fase apenas ocorre se a língua for considerada portuguesa e consiste em tentar provar que é português do Brasil, caso não sejam encontrados indícios é considerada português de Portugal. Nesta fase as línguas asiáticas já foram colocadas de parte devido às diferenças e à inexistência de letras de a-z resultando no valor máximo no cálculo da distância. É possível dividir os *tweets* em palavras e em N-grams. Nesta fase é verificada a existência de:

- Trigramas *kkk* típicos do "rir" os brasileiros.
- Trigramas *rsr* típicos do "rir" dos brasileiros.
- Gerúndio, trigramas *ndo* no final das palavras, excluindo palavras usuais do português que terminam em *ndo* e não são verbos;
- Palavras de português do Brasil (ex: "ônibus" em português de Portugal autocarro). Foi criada uma *stop list* de cerca de setenta palavras.

Para cada um das quatros "pistas" acima indicadas é atribuído um valor, o resultado final resulta na soma dos quatro valores. Se este valor for significativo os *tweets* são considerados como português do Brasil. A avaliação deste método encontra-se presente na secção 5.2.2.

3.4.6 Verificação de contas

No Twitter um utilizador pode apagar a conta ou ter esta suspensa por infracção das regras. Detectar estes casos é de extrema importância devido ao processo de escalonamento em uso (ver secção 3.3.4). Os utilizadores que não recebem dados há mais tempo são os mais prioritários na sua classe, e pedidos de dados de utilizadores com contas suspensas ou inexistentes não retornam dados (secção A.1.3). Isto leva a que os utilizadores inválidos se acumulem no topo das classes de prioridades, desperdiçando importantes recursos. Numa situação limite, ao final de um período longo de tempo, vão ser acumulados tantos utilizadores inválidos que apenas estes são verificados, levando à situação em que não são obtidos mais *tweets*.

Verificar se um utilizador possui conta válida requer aceder à página do Twitter e analisar o conteúdo desta ou pedir informações sobre o utilizador em questão à REST API. A primeira opção é mais pesada em termos computacionais e a segunda o número de pedidos é limitado (ver secção A.4).

A questão que se coloca é que utilizadores devem ser verificados e com que frequência. A situação ideal é: quando a conta de um utilizador se torna inválida no Twitter quase em tempo real é detectado e assinalado como tal na BD. Isto exigiria verificar milhares de utilizadores todos os minutos, criando problemas de escalabilidade. Como referi anteriormente o Twitter não devolve dados de contas inválida, invés de serem verificados todos os utilizadores periodicamente, podem ser verificados apenas os utilizadores cujo TwitterEcho não recebe dados há mais tempo. Desta forma o problema de verificação é reduzido de milhares de utilizadores para algumas dezenas. A verificação é efectuada acedendo à página do Twitter.

3.4.7 "Congelamento" de inactivos

Na presente data o TwitterEcho recolhe dados de cerca de 60 mil utilizadores, aproximadamente 60% não publicam há mais de um mês. Um longo período de inactividade indicia que o utilizador deixou de utilizar o Twitter ou usa este apenas para obter informações.

Não existe nenhuma garantia que um utilizador que deixou de publicar por um longo período de tempo não voltará a publicar, no entanto, enviar estes utilizadores frequentemente para os clientes gera um desperdício de recursos. Os utilizadores inactivos apenas precisam de ser verificados esporadicamente. Estes utilizados são verificados no período das 3 às 7 da manhã, porque neste período a actividade dos utilizadores é reduzida (ver figura 3.9). Devido à actividade de utilizadores ser reduzida, verificar um maior número de utilizadores neste período não tem impacto negativo significativo.

É necessário escolher os utilizadores que devem ser verificados apenas no período das 3 às 7 da manhã. Espera-se que um utilizador que não publica há quinze dias terá uma

maior probabilidade de não voltar a publicar que um utilizador que não publica há três dias. Os utilizadores escolhidos para serem "congelados" deverão ser utilizadores com uma probabilidade baixa de voltarem a publicar. Desenvolvi um algoritmo, que usa os dados presentes na BD (mais de 4 milhões de estatísticas), e quantifica quantos utilizadores voltaram a publicar após X dias inactivos, sendo X entre 1 a 40 dias de inactividade. O número de utilizadores que voltaram a publicar após um período inactivo de 1 a 40 dias encontra-se ilustrado na figura 3.24.

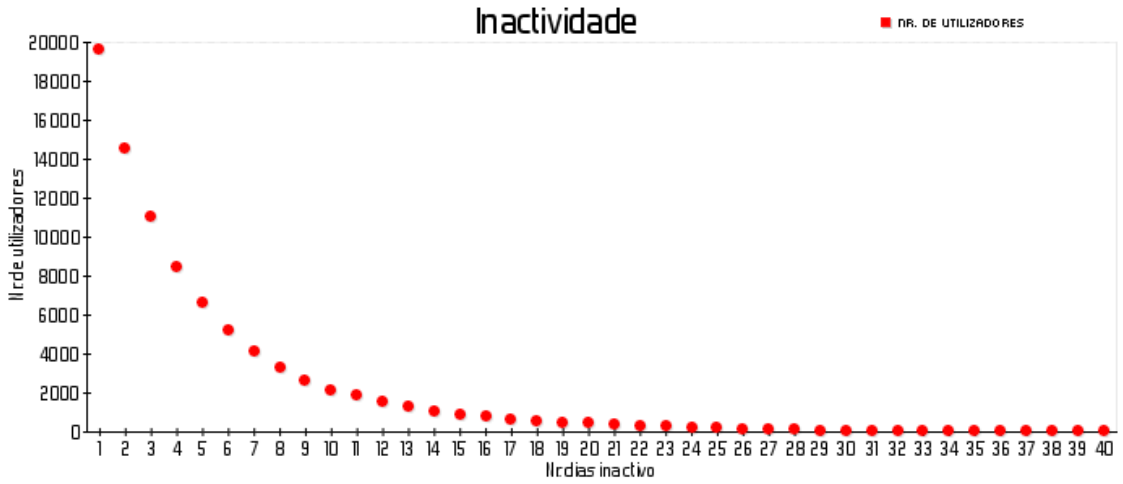


Figura 3.24: Número de utilizadores que voltaram a publicar após X dias inactividade. O eixo x representa o número de dias de inactividade e o eixo Y o número de utilizadores que voltaram a publicar.

É possível verificar na figura 3.24 que como esperado o número de utilizadores que volta a publicar tende gradualmente para zero com o aumento do número de dias inactivos. O número de utilizadores que publicaram de novo após mais de duas semanas de inactividades encontra-se ilustrado na figura 3.25.

É possível concluir a partir da figura 3.25 que a partir de vinte e cinco dias inactivos o número de utilizadores que volta a publicar é pouco significativo (aproximadamente 100). Com base nestes dados escolhi "congelar" utilizadores que não publicam há mais de um mês. Numa comunidade de dimensão superior caso não se encontrem disponíveis um grande número de máquinas para obter informações este número pode ser reduzido para aproximadamente quinze dias com um impacto pouco significativo na perda de *tweets*.

3.4.8 Anotação do crescimento da BD

Este módulo quando executado grava o número de registos de todas as tabelas da BD. Isto permite de forma fácil monitorizar a evolução do conjunto de dados recolhidos.

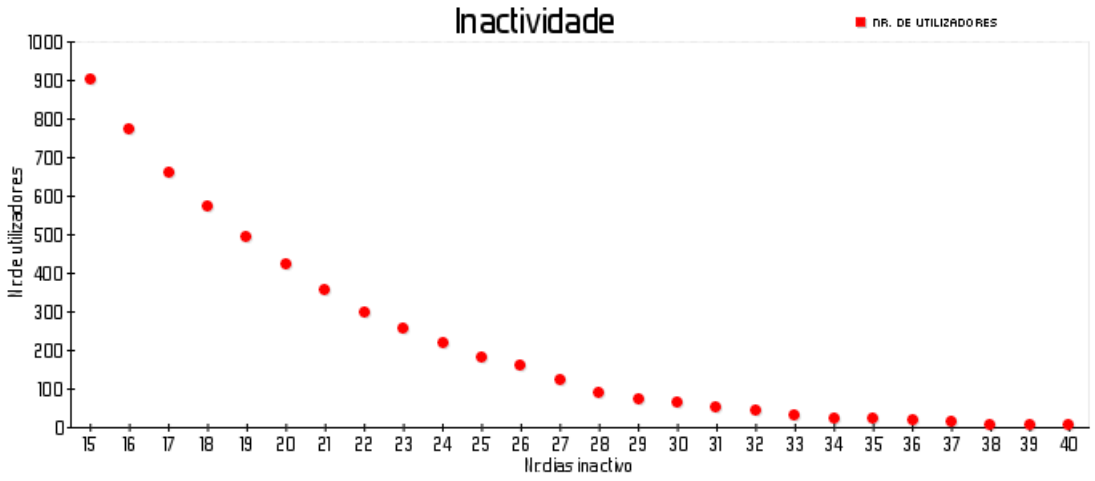


Figura 3.25: Número de utilizadores que voltaram a publicar após X dias inactividade. O eixo x representa o número de dias de inactividade e o eixo Y o número de utilizadores que voltaram a publicar. Este gráfico é uma parte do gráfico da figura 3.24.

3.5 Crescimento da BD

O TwitterEcho foi colocado em funcionamento em 27 de Abril de 2011 às 14:40. A BD continha 2052 utilizadores portugueses. Em 24h foram adicionados de forma autónoma pelo sistema 4017 utilizadores. Isto representa um crescimento de aproximadamente 200% em 24h face à amostra inicial.

No período de 27 de Abril a 17 de Junho foram adicionados 65225 utilizadores e recolhidos 2832473 *tweets*, 215645 listas de seguidores, 130373 listas de amigos e 4502387 estatísticas.

Na tabela 3.2 encontram-se representados os dados obtidos em média em função de diferentes períodos de tempo.

	Dia	**Hora**	**Minuto**	**Segundo**
Utilizadores	1269	53	0.88	0.01
Tweets	55118	2297	38.28	0.64
Seguidores	4196	175	2.91	0.05
Amigos	2537	106	1.76	0.03
Estatísticas	87614	3651	60.84	1.01
Total	150735	6281	104.68	1.74

Tabela 3.2: Número de dados obtidos médios.

Os valores acima indicados são valores médios, o sistema consegue lidar com números bem mais elevados. A recolha de *tweets* não é uniforme no tempo. Encontra-se dependente da actividade dos utilizadores. Por exemplo, no dia das eleições legislativas 5

de Junho o TwitterEcho recolheu 77451 *tweets*, o maior número desde sempre. O número máximo de *tweets* inseridos na BD num minuto foram 612 e num segundo 149.

O TwitterEcho ainda possui margem para crescimento do número de utilizadores a obter informações, não tendo chegado a um ponto de estagnamento, por exemplo foram adicionados 825 novos utilizadores no dia 17 de Junho.

Capítulo 4

Interface *Web*

O TwitterEcho é um sistema complexo que é afectado por inúmeras variáveis: Twitter, Twitter API, clientes, BD, rede, entre outras. A ocorrência de uma falha que impossibilite a recolha de *tweets*, significa uma perda significativa de dados e um "vazio" nesse período temporal. Torna-se portanto, necessária a existência de uma aplicação que permita monitorizar o TwitterEcho.

O TwitterEcho armazena na BD erros e dados importantes para detecção de falhas. Além disso os módulos sempre que são executados guardam um sumário da execução em ficheiros. Devido à quantidade de informação disponível, torna-se ineficiente visualizar estes dados em bruto. É necessário tratá-los e agrupá-los para visualização.

Desenvolvi uma aplicação denominada TwitterEcho *Web* para monitorização do TwitterEcho. No desenvolvimento utilizei HTML, CSS, JavaScript e PHP. Uso também uma biblioteca GD de PHP para gerar imagens [PHP] e um sistema desenvolvido por mim que permite efectuar cache de páginas.

O sistema de cache é executado periodicamente para actualizar e guardar em disco páginas que demorem a serem geradas, por usarem instruções SQL pesadas. Sendo estas disponibilizadas directamente ao utilizador quando requisitadas. Estas páginas são a maioria das vezes constituídas por estatísticas, médias, entre outros dados, existindo diferenças mínimas entre uma versão actual e uma versão de uma hora atrás. O utilizador possui a opção de requisitar uma versão actualizada de uma página em cache, esta será guardada na sessão do utilizador. Sempre que um utilizador visite uma página que se encontra guardada em sessão e esta seja mais recente que a guardada em disco, será esta a disponibilizada.

No desenvolvimento desta aplicação primei pela simplicidade de *design*, respeito pelas normas W3C [W3C], facilidade de aprendizagem e organização de informação. A

organização de informação, de forma eficiente, foi a tarefa mais complexa de realizar, devido à quantidade de dados enorme a disponibilizar. A página inicial é:

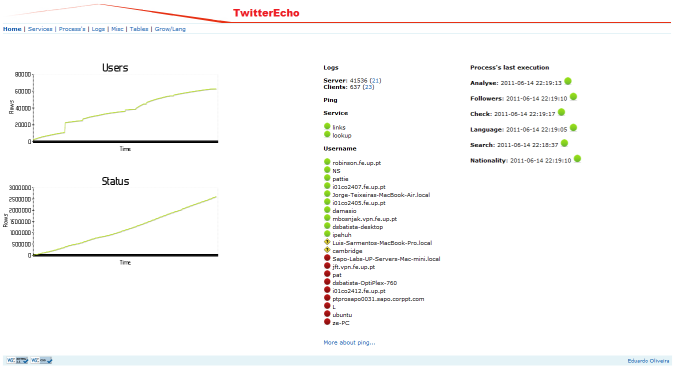


Figura 4.1: Página inicial do TwitterEcho *Web*.

A página inicial permite visualizar o crescimento da BD (gráficos do lado esquerdo), o estado dos clientes (coluna central) e dos módulos (coluna da direita). Optei por usar gráficos sempre que possível para tornar a aplicação atractiva e porque estes permitem agrupar informação facilmente.

O TwitterEcho *Web* permite monitorizar: máquina, MySQL, clientes e serviços, módulos, erros e base de dados.

4.1 Máquina

A aplicação permite consultar os *loads* [fU] da máquina, processos que o Apache tem em execução e módulos em execução. Isto permite que um utilizador se aperceba do estado da máquina e detecte sobrecargas. Exemplo dos processos a correr num dado momento:

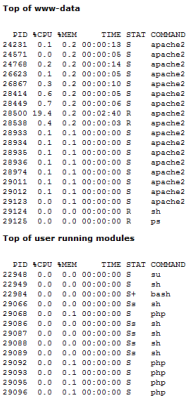
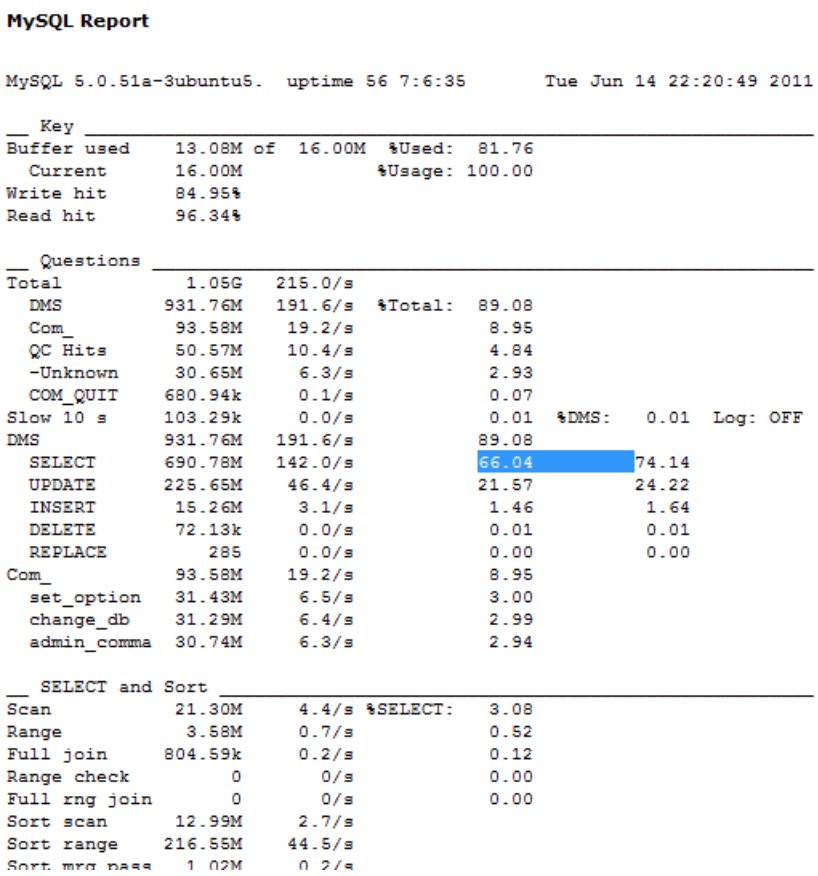


Figura 4.2: Processos em execução no Apache e módulos.

4.2 MySQL

A aplicação permite consultar estatísticas do MySQL, como por exemplo número médio de pesquisas, inserções e actualizações por segundo. A *performance* do TwitterEcho está intimamente relacionada com o MySQL. Parte das estatísticas podem ser observadas na seguinte figura:



MySQL Report

```
MySQL 5.0.51a-3ubuntu5.  uptime 56 7:6:35       Tue Jun 14 22:20:49 2011

__ Key _________________________________________________________________
Buffer used    13.08M of  16.00M  %Used:  81.76
  Current      16.00M             %Usage: 100.00
Write hit      84.95%
Read hit       96.34%

__ Questions ___________________________________________________________
Total           1.05G   215.0/s
  DMS         931.76M   191.6/s  %Total:  89.08
  Com_         93.58M    19.2/s            8.95
  QC Hits      50.57M    10.4/s            4.84
  -Unknown     30.65M     6.3/s            2.93
  COM_QUIT    680.94k     0.1/s            0.07
Slow 10 s     103.29k     0.0/s            0.01  %DMS:   0.01  Log: OFF
DMS           931.76M   191.6/s           89.08
  SELECT      690.78M   142.0/s           66.04          74.14
  UPDATE      225.65M    46.4/s           21.57          24.22
  INSERT       15.26M     3.1/s            1.46           1.64
  DELETE       72.13k     0.0/s            0.01           0.01
  REPLACE         285     0.0/s            0.00           0.00
Com_           93.58M    19.2/s            8.95
  set_option   31.43M     6.5/s            3.00
  change_db    31.29M     6.4/s            2.99
  admin_comma  30.74M     6.3/s            2.94

__ SELECT and Sort _____________________________________________________
Scan           21.30M     4.4/s %SELECT:   3.08
Range           3.58M     0.7/s            0.52
Full join     804.59k     0.2/s            0.12
Range check         0       0/s            0.00
Full rng join       0       0/s            0.00
Sort scan      12.99M     2.7/s
Sort range    216.55M    44.5/s
Sort mrg pass   1.02M     0.2/s
```

Figura 4.3: Estatísticas do MySQL.

4.3 Clientes e serviços

Num sistema distribuído é fundamental possuir informação sobre que máquinas no actual momento enviam dados. O TwitterEcho armazena essa informação, sendo disponibilizada na interface *Web* (figura 4.4).

No caso dos serviços disponibiliza o tempo de execução destes, médias, máximos, mínimos, últimas execuções, entre outros. Por exemplo é possível visualizar os tempos de execução dos últimos serviços na figura 4.5.

More

Username	Service	Last Date
cambridge	links	2011-06-14 20:22:03
	lookup	2011-06-14 20:22:59
damasio	links	2011-06-14 22:12:03
	lookup	2011-05-21 20:14:17
dsbatista-desktop	links	2011-06-14 22:21:57
dsbatista-OptiPlex-760	links	2011-06-10 16:51:54
	lookup	2011-06-02 15:04:49
i01co2405.fe.up.pt	links	2011-06-14 22:12:59
	lookup	2011-06-14 22:21:54
i01co2407.fe.up.pt	links	2011-04-27 19:56:47
	lookup	2011-06-14 22:21:54
i01co2412.fe.up.pt	links	2011-05-09 19:01:41
	lookup	2011-06-03 15:51:58
ipehuh	links	2011-06-14 22:20:48
jft.vpn.fe.up.pt	lookup	2011-06-13 22:41:17
Jorge-Teixeiras-MacBook-Air.local	lookup	2011-06-14 22:17:00
L	lookup	2011-05-25 15:22:12
Luis-Sarmentos-MacBook-Pro.local	links	2011-06-14 20:22:01
	lookup	2011-06-14 20:30:48
mbosnjak.vpn.fe.up.pt	lookup	2011-06-14 22:21:54
NS	links	2011-06-11 14:36:07
	lookup	2011-06-14 22:21:56

Figura 4.4: Parte da lista de máquinas que enviam dados para o servidor e última data de envio.

Home | **Services** | Process's | Logs | Misc | Tables | Grow/Lang

Running Time
Averages
Percentages

Sample
Head
Tail

Tail

Service	Date	Time (s)
Lookup Get	2011-06-14 22:23:45	0.114087
Lookup Put	2011-06-14 22:23:45	0.12427
Lookup Get	2011-06-14 22:23:44	0.116818
Lookup Get	2011-06-14 22:23:43	0.134782
Lookup Put	2011-06-14 22:23:43	0.14715
Lookup Get	2011-06-14 22:23:42	0.110396
Lookup Put	2011-06-14 22:23:42	0.169073
Lookup Put	2011-06-14 22:23:41	0.153483
Lookup Get	2011-06-14 22:23:40	0.116061
Lookup Put	2011-06-14 22:23:40	0.141369
Lookup Put	2011-06-14 22:23:40	0.183852
Lookup Get	2011-06-14 22:23:36	0.110048
Lookup Put	2011-06-14 22:23:36	0.170975
Lookup Get	2011-06-14 22:23:35	0.131221
Lookup Get	2011-06-14 22:23:35	0.109482
Lookup Put	2011-06-14 22:23:35	0.159359
Lookup Get	2011-06-14 22:23:35	0.116053
Lookup Put	2011-06-14 22:23:35	0.850897
Lookup Get	2011-06-14 22:23:33	0.111011
Lookup Put	2011-06-14 22:23:33	0.153993

Figura 4.5: Tempo de execução dos últimos pedidos aos serviços.

4.4 Módulos

A interface *Web* disponibiliza um conjunto variado de informações sobre os módulos, na página inicial (figura 4.1 lado direito) disponibiliza o estado destes. Se qualquer módulo não se encontrar no *crontab* ou não executar correctamente aparecerá assinalado. Disponibiliza para cada módulo os sumários destes, permitindo visualizar por exemplo as últimas execuções:

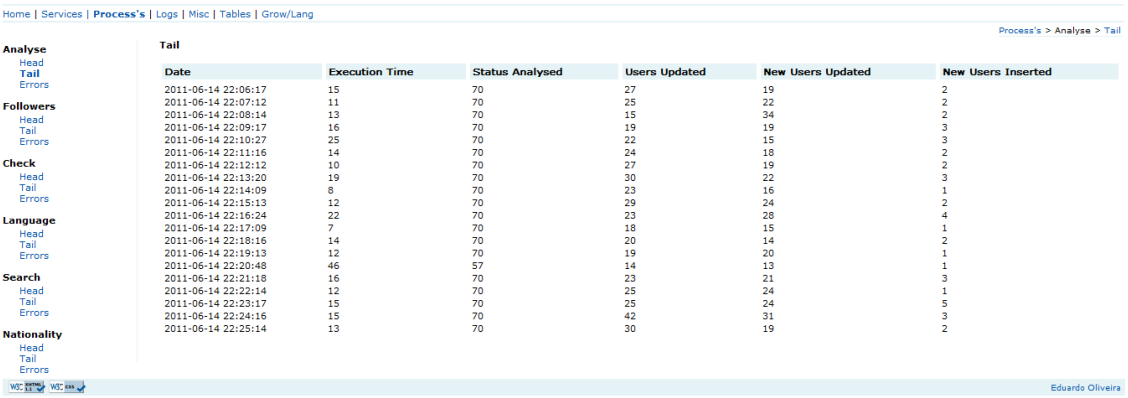


Figura 4.6: Últimas execuções do módulo análise de *tweets*.

Possui também estatísticas distintas para cada módulo. Por exemplo para o módulo de análise de *tweets*, cada execução em média demora onze segundos, são analisados trinta e oito *tweets* e são inseridos nove possíveis utilizadores. Cria também gráficos, por exemplo a figura 3.11 foi criada pela interface *Web*.

4.5 Erros

Os erros encontram-se divididos em dois tipos: detectados pelo servidor e detectados pelo cliente. Os erros detectados pelos clientes podem ser erros gerados por si (por exemplo perda de acesso ao Twitter) ou gerados pelo servidor (por exemplo um cliente pede uma lista de utilizadores ao servidor e ao invés de utilizadores recebe algo inesperado, o cliente reporta ao servidor).

O TwitterEcho *Web* no caso dos erros detectados pelo servidor permite:

- Visualizar número de erros.
- Visualizar tipos de erro distintos (nome, data da última ocorrência e número de ocorrências).
- Visualizar últimos erros.
- Visualizar últimos erros por tipo.

- Efectuar pesquisas, figura 4.7.

No caso de erros detectados por clientes permite:

- Visualizar número de erros.
- Visualizar estatísticas por cada computador que execute clientes (data de último erro, número de erros e percentagem de erros face ao total de todos os computadores que executem clientes).
- Visualizar estatísticas iguais às anteriores por cada cliente (lookup ou links).
- Visualizar tipos de erro distintos (nome, data da última ocorrência e número de ocorrências).
- Visualizar últimos erros por computador.
- Efectuar pesquisas, figura 4.7.

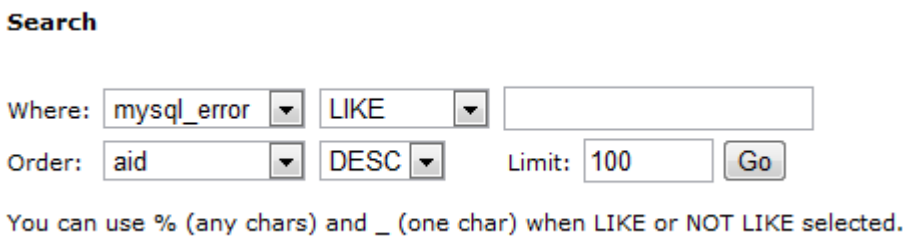


Figura 4.7: Formulário para pesquisa.

4.6 Base de Dados

O TwitterEcho *Web* permite consultar os dados da BD, além disso permite consultar espaço ocupado (figura 4.8), a estrutura (figura 4.9), crescimento da BD (figura 4.10), entre outros.

Para cada tabela utilizadores, *tweets*, estatísticas, seguidores e amigos é possível visualizar as últimas inserções, efectuar pesquisas (figura 4.7), visualizar estatísticas. Exemplo da página principal de visualização de dados na figura 4.11.

As estatísticas geradas são distintas consoante a tabela. Utilizadores possui informações sobre: número utilizadores activos, média de menções e *retweets*, média de prioridades (lookup e links), datas variadas, média de atraso entre *get's* e *put's*. Sobre os *tweets* disponibiliza: número de *tweets*, média de *tweets* por utilizador, tempo médio em que um utilizador recebe um tweet após ser inserido, tempo médio entre o *tweet* ser criado e inserido, percentagem de utilizadores com *tweets*, lista de utilizadores mais activos, lista de

Table	Rows	Average Row KB	Size MB	Indexs MB	Total MB	Auto Increment	Last Update
followers	208411	5.99	1218.25	4.43	1222.68	208412	2011-06-14 22:27:05
friends	125467	4.59	562.36	2.65	565.01	125468	2011-06-14 22:27:05
lang	6636	0.02	0.11	0.08	0.19		2011-05-25 00:25:02
log	41538	0.68	27.68	0.41	28.09	41539	2011-06-14 22:22:17
log_clients	637	0.09	0.06	0.00	0.06		2011-06-14 15:59:54
log_debug	72	0.98	0.07	0.00	0.07	73	2011-05-26 18:14:34
log_ping	33	0.03	0.00	0.00	0.00		2011-06-14 22:27:55
log_tables	21336	0.02	0.49	0.12	0.61		2011-06-14 22:00:01
log_time	740596	0.01	7.06	13.28	20.34		2011-06-14 22:27:55
log_web	2286	0.03	0.07	0.05	0.12	2287	2011-06-12 12:06:51
new_users	709167	0.25	173.58	29.91	203.49		2011-06-14 22:27:55
other_countrys_citys	187	0.02	0.00	0.01	0.01		2011-06-02 16:39:34
other_names	309	0.02	0.01	0.01	0.01		2011-04-27 14:39:40
portuguese_citys	192	0.02	0.00	0.01	0.01		2011-05-10 18:31:44
statistics	4171972	0.02	99.47	89.47	188.94	4171973	2011-06-14 22:27:56
statistics_by_user_may_diff	37251	0.04	1.42	0.00	1.42		2011-06-02 22:27:51
statistics_by_user_may_last	37251	0.02	0.89	0.00	0.89		2011-06-02 19:15:18
status	2617485	0.21	524.32	99.07	623.38	2617486	2011-06-14 22:27:55
status_temp	6377470	0.10	621.58	179.56	801.13		2011-06-14 22:27:35
users	62660	0.22	14.30	3.30	17.60		2011-06-14 22:27:55

Figura 4.8: BD.

Struct of status

Field	Type	Null	Key	Default	Extra
aid	int(15)	NO	MUL		auto_increment
status_id	bigint(25)	NO	PRI		
user_id	int(15)	NO	MUL		
hash	char(32)	NO			
checked_mentions_retweets	tinyint(1)	NO		0	
date_insert	timestamp	NO		CURRENT_TIMESTAMP	
created_at	timestamp	NO		0000-00-00 00:00:00	
text	tinytext	NO			
source	tinytext	YES			
truncated	tinyint(1)	NO			
in_reply_to_status_id	bigint(25)	YES		0	
in_reply_to_user_id	int(15)	YES		0	
in_reply_to_screen_name	varchar(32)	YES			
retweet_count	int(11)	YES		0	

Indexs

Key_name	Seq_in_index	Column_name	Cardinality
PRIMARY	1	status_id	2617586
user_id	1	user_id	
aid	1	aid	

Figura 4.9: Estrutura da tabela *status*.

utilizadores com mais respostas e *tweets* com mais respostas. Os dados disponibilizados sobre as estatísticas são: número de estatísticas, média por utilizador, percentagem de utilizadores com estatísticas e máximo, mínimo e média do número de *tweets*, seguidores e amigos. Nos seguidores e amigos é possível visualizar a percentagem de utilizadores com seguidores/amigos. Para os utilizadores, *tweets*, seguidores e amigos também é possível visualizar gráficos com o número de inserções ao longo do dia, mês, e desde sempre (em várias escalas hora, dia e mês). Exemplo do número de *tweets* inseridos no mês de Junho por hora na figura 4.12.

Sobre os utilizadores é disponibilizada uma página com informação sobre as prioridades do serviço lookup, percentagem de utilizador de cada classe e os tempos da última

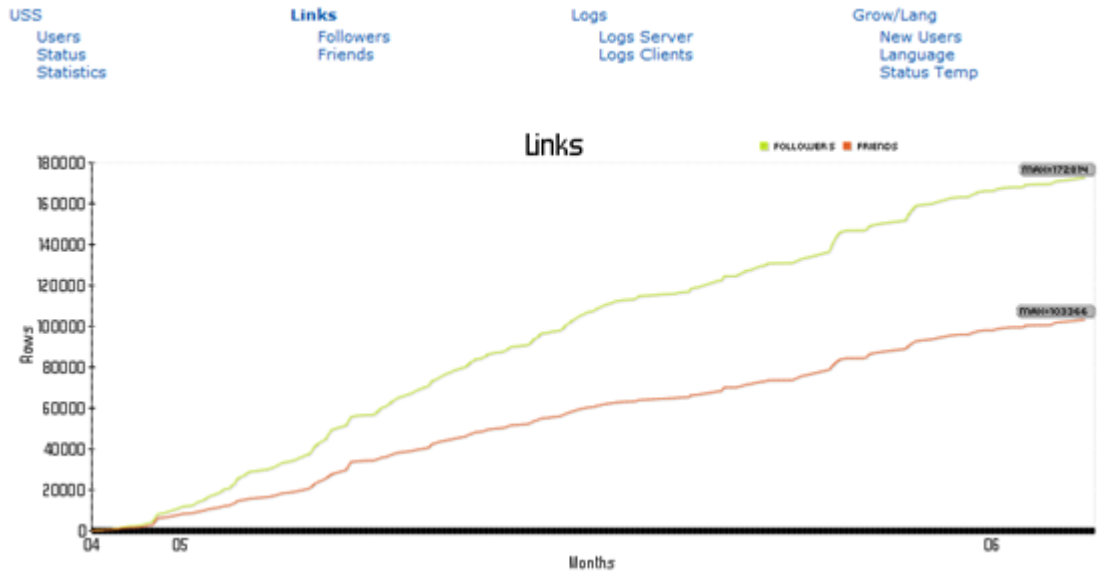


Figura 4.10: Crescimento das tabelas de relações ao longo do tempo.

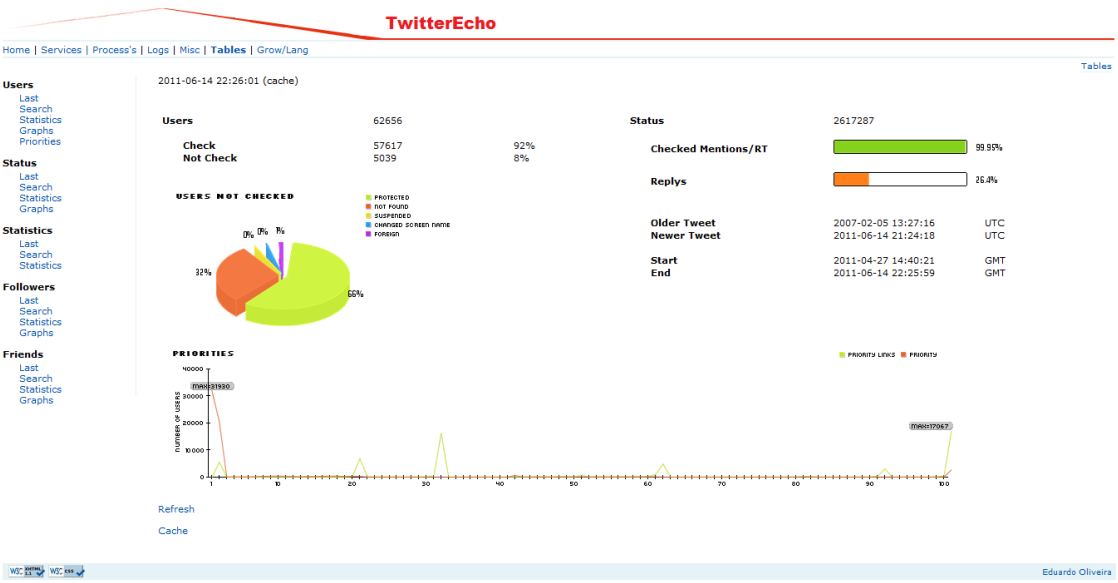


Figura 4.11: Página principal de visualização de dados.

actualização para cada classe (figura 4.13).

O TwitterEcho *Web* permite também obter informações dos possíveis utilizadores, das avaliações de linguagem realizadas e dos *tweets* recolhidos para essa avaliação.

Sobre os possíveis utilizadores permite pesquisar e visualizar: os últimos inseridos, os utilizadores com mais menções ou *retweets* que foram marcados como não portugueses,

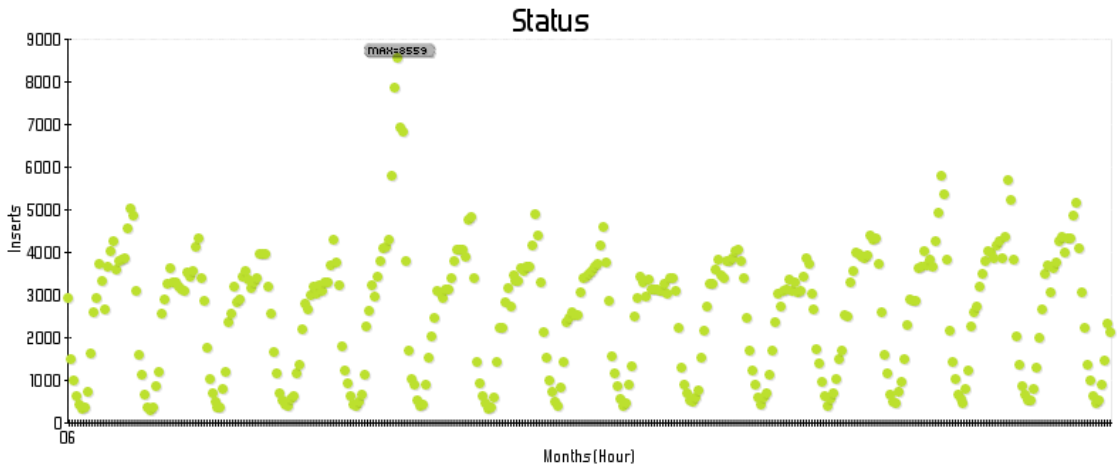


Figura 4.12: Número de *tweets* inseridos por hora no mês de Junho (os períodos com poucas inserções corresponde ao período das 3 as 8 da manha). E os picos no dia 5 de Junho estão relacionados com as eleições legislativas.

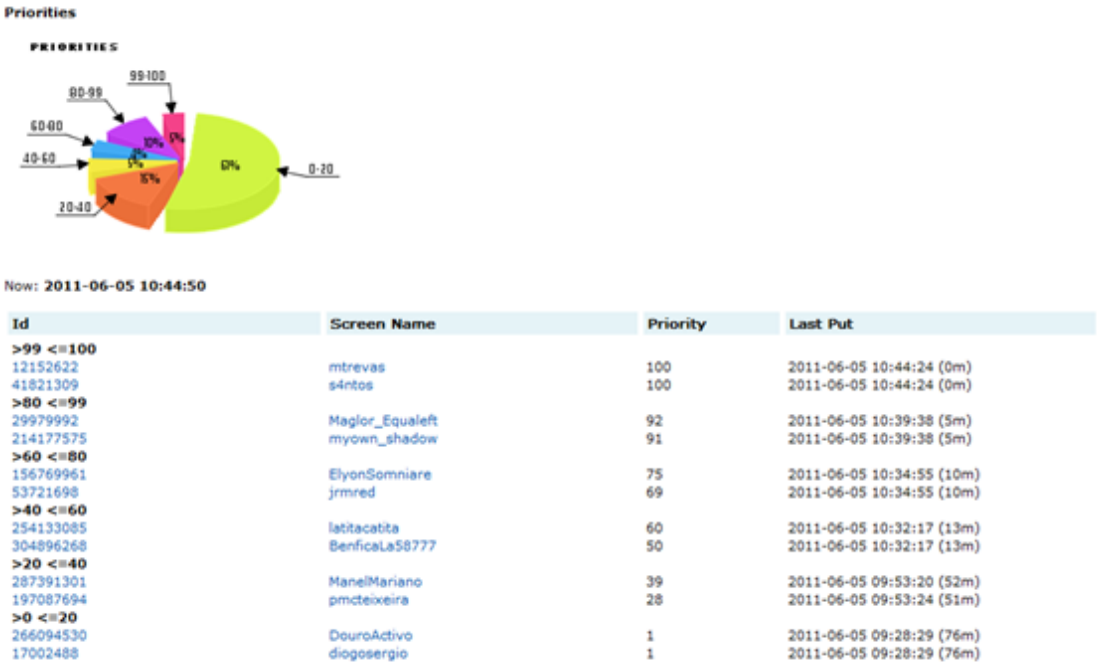


Figura 4.13: Prioridades dos utilizadores para o serviço lookup.

os utilizadores com mais menções ou *retweets* cuja identificação de nacionalidade não foi conclusiva, um utilizador aleatório cuja identificação de nacionalidade não foi conclusiva. É possível adicionar utilizadores que foram assinalados como não portugueses ou que a análise foi inconclusiva manualmente através do TwitterEcho *Web*, sendo essa acção registada.

Referente a análise de linguagem permite pesquisar e visualizar: últimas análises, últimas que resultaram em identificação como português e últimas que a identificação de linguagem ficou muito próxima de identificar como português. É possível visualizar os *tweets* que deram origem a qualquer decisão de linguagem.

No presente capítulo expliquei abreviadamente as principais funcionalidades, no entanto a aplicação *Web* devido à sua dimensão[1] não é possível no contexto do presente documento entrar em detalhes sobre todas as funcionalidades. É possível experimentar uma demo da aplicação [2].

[1] Reflecte a complexidade do TwitterEcho

[2] http://robinson.fe.up.pt/twitter/admin2-demo/ (disponível entre 1 e 15 de Julho de 2011)

Capítulo 5

Avaliação

No presente capítulo descrevo primeiramente a metodologia de avaliação que emprego e depois os resultados. Com base nos resultados enuncio trabalho futuro.

5.1 Metodologia

5.1.1 Nacionalidade de utilizadores

O TwitterEcho identifica a nacionalidade dos utilizadores em três grupos:

- Utilizadores portugueses;
- Utilizadores não portugueses;
- Utilizadores cuja nacionalidade é desconhecida (ainda não obteve dados suficientes para efectuar uma decisão).

Para cada grupo selecciono uma amostra aleatoriamente e tomo uma decisão manual sobre cada utilizador. A amostra seleccionada não contém utilizadores adicionados manualmente através da interface *Web*. Para tomar a decisão visualizo o perfil do utilizador, *tweets* e listas de amigos e seguidores. Para cada grupo anoto os resultados das avaliações a cada cem utilizadores parando quando considero que a amostra é significativo e os resultados estáveis.

5.1.2 Identificação de linguagem

O meu foco é avaliar a correcta identificação de português de Portugal. Esta avaliação deve ser efectuada para conjuntos de *tweets* de diferentes dimensões. Para efectuar esta avaliação necessito de garantir que a amostra seleccionada possui as seguintes características:

- Contém um número significativo de *tweets* em português de Portugal.
- Contém *tweets* em várias línguas.
- A dimensão do conjunto de *tweets* a avaliar varia.

O TwitterEcho possui na presente data mais de seis milhões de *tweets* de utilizadores cuja decisão do módulo de perfil foi inconclusiva e dois milhões e meio de utilizadores marcados como portugueses. De forma a garantir que a amostra possui um número significativo de *tweets* em português e possui *tweets* em outras línguas selecciono a amostra a partir dos dois conjuntos de *tweets*.

O algoritmo que gera a amostra e visa garantir as características pretendidas anteriormente mencionadas, é o seguinte:

1. Escolher um utilizador marcado como portugueses ou nacionalidade desconhecida.
2. Obter um número aleatório entre 30 e 100 de *tweets* desse utilizador.
3. Os *tweets* são pré-processados, excluindo menções, URL's, entre outros (o algoritmo usado para pré-processamento é o mesmo usado no módulo de identificação de linguagem, secção 3.4.5).
4. O texto após pré-processamento é mostrado e é dado a escolher um conjunto de opções.

No ponto 1 é seleccionado um número aleatório entre 0 e 3, caso seja 0 é escolhido um utilizador de nacionalidade desconhecida. Entre 1 e 3 é escolhido um utilizador marcado como português.

As opções dadas a escolher no ponto 4 são:

- Português de Portugal, os *tweets* encontram-se escritos em português de Portugal podendo conter palavras isoladas noutra língua.
- Outra língua/s, os *tweets* encontram-se escritos noutra/s língua/s que não português de Portugal.
- Várias línguas, estando português de Portugal presente e é predominante.
- Várias línguas, estando português de Portugal presente, mas não é predominante.

Avalio os resultados com base na precisão, abrangência, taxa de negativos verdadeiros e exactidão (informações sobre estas medidas de desempenho no apêndice C).

5.1.3 Perda de *tweets*

Nas avaliações anteriores o foco é em provar que o TwitterEcho recolhe dados de utilizadores portugueses. Nesta avaliação o meu foco é provar que o TwitterEcho recolhe uma quantidade significativa de *tweets* publicados pelos utilizadores. Para esse efeito, determino para uma amostra significativa de utilizadores o número de *tweets* presentes na BD em função do número de *tweets* do utilizador.

Desenvolvi uma aplicação que para um conjunto de utilizadores aleatórios presentes na BD, obtém do Twitter até duzentos *tweets* recentes de cada utilizador (método *Statuses/user_timeline*da REST API secção A.1.2). Para cada utilizador verifica quantos destes *tweets* se encontram na BD.

5.2 Resultados

5.2.1 Nacionalidade de utilizadores

Para os utilizadores marcados como portugueses seleccionei uma amostra aleatória de 1000 utilizadores de cerca de 60 mil. Anotei as avaliações a cada cem utilizadores. Avaliei manualmente 500 utilizadores, os resultados encontram-se presentes na seguinte tabela:

Número de utilizadores avaliados	**Número de utilizadores assinalados manualmente como portugueses**	**Percentagem de utilizadores marcados na BD como portugueses e assinalados manualmente de acordo**
100	93	93
200	188	94
300	276	92
400	363	90,75
500	454	90,8

Tabela 5.1: Resultados da avaliação dos utilizadores marcados como portugueses.

É possível concluir que o TwitterEcho está a recolher informações maioritariamente de utilizadores portugueses. Extrapolando estes resultados é possível estimar que dos sessenta mil utilizadores aproximadamente 55 mil são portugueses.

Procedi de forma similar para classificar utilizadores marcados como não portugueses. Avaliei 800 utilizadores de cerca de meio milhão. Os resultados estão presentes na tabela 5.2.

Número de utilizadores avaliados	**Número de utilizadores assinalados manualmente como não portugueses**	**Percentagem de utilizadores marcados na BD como não portugueses e assinalados manualmente de acordo**
100	100	100
200	200	100
300	300	100
400	398	99,5
500	497	99,5
600	596	99,3
700	696	99,4
800	796	99,5

Tabela 5.2: Resultados da avaliação dos utilizadores marcados como não portugueses.

É possível verificar através dos resultados presentes na tabela 5.2 que a esmagadora maioria dos utilizadores são assinalados correctamente como não portugueses. Extrapolando estes resultados é possível estimar que cerca de 2500 utilizadores foram classificados como não portugueses não o sendo. Comparando este valor com a estimativa de 55 mil utilizadores portugueses presentes na BD, este valor representaria um incremento de cerca de 4,5% de utilizadores.

Procedi de forma similar para classificar os utilizadores cuja nacionalidade é desconhecida. Avaliei 500 utilizadores de cerca de 170 mil. Os resultados estão presentes na seguinte tabela:

Número de utilizadores avaliados	**Número de utilizadores assinalados manualmente como portugueses**	**Percentagem de utilizadores marcados na BD como nacionalidade desconhecida que são portugueses**
100	14	14
200	27	13,5
300	38	12,7
400	50	12,5
500	60	12

Tabela 5.3: Resultados da avaliação dos utilizadores marcados com nacionalidade desconhecida.

É possível concluir através destes resultados que existe um número significativo de portugueses no conjunto de utilizadores marcados como nacionalidade desconhecida. Extrapolando estes resultados é possível estimar que cerca de 20 mil utilizadores são portugueses, este valor representaria um aumento significativo de 36% dos utilizadores portugueses presentes na BD.

Durante a avaliação dos utilizadores cuja nacionalidade é desconhecida tive que recorrer várias vezes à lista de amigos e seguidores, devido a possuírem poucos ou nenhuns *tweets*. De futuro pode ser implementado um método que caso a análise de perfil e linguagem não seja suficiente decida a nacionalidade com base nos amigos e seguidores de um utilizador.

Os utilizadores marcados como portugueses que na realidade não o são, grande parte destes são brasileiros residentes em Portugal. De futuro para reduzir estes casos, o método de identificação de perfil deverá trabalhar em conjunto com o módulo de identificação de linguagem.

5.2.2 Identificação de linguagem

Efectuei 550 avaliações. Nestas avaliações encontrei *tweets* em várias línguas, como por exemplo: português (de Portugal e Brasil), espanhol, inglês, francês, italiano, alemão, línguas asiáticas, entre outras. Os resultados das avaliações encontram-se presentes na seguinte tabela (informações sobre as medidas de desempenho no apêndice C):

Nr. de *Tweets*	**Avaliações**	**TP**	**FP**	**FN**	**TN**	**Precisão**	***Recall***	**Negativos**	**Exactidão**
30-40	115	24	1	25	65	0,96	0,49	0,985	0,774
41-60	193	57	0	29	107	1	0,663	1	0,85
61-80	117	37	2	12	66	0,949	0,755	0,971	0,88
81-100	125	31	1	11	82	0,969	0,738	0,988	0,904
30-100	550	149	4	77	320	0,974	0,659	0,988	0,853

Tabela 5.4: Resultados da identificação de linguagem.

É possível verificar que a identificação da língua portuguesa de Portugal possui uma precisão elevada e não se encontra relacionada com o número de *tweets*. No entanto, a abrangência (*recall*) encontra-se directamente relacionada com o número de *tweets* disponíveis para avaliação. Isto deve-se ao facto, que um texto em português de maior dimensão tem maior probabilidade de se aproximar do perfil em uso para comparação, que um texto curto.

Durante o processo de avaliação manual classifiquei vários conjuntos de *tweets* como contendo várias línguas, retirando essas avaliações os resultados são:

É possível verificar que a abrangência média aumentou em seis pontos percentuais, isto deve-se à diminuição de falsos negativos. O módulo de identificação de linguagem não possui mecanismos para lidar com a presença de várias línguas. No entanto, a presença de várias línguas não afecta a precisão.

A grande maioria das vezes em que além do português existe outra língua presente, esta é o inglês. Como trabalho futuro seria interessante identificar pelo menos este caso

Nr de. *Tweets*	**Avaliações**	**TP**	**FP**	**FN**	**TN**	**Precisão**	***Recall***	**Negativos**	**Exactidão**
30-40	104	24	1	19	60	0,96	0,558	0,984	0,808
41-60	183	56	0	23	104	1	0,709	1	0,874
61-80	107	36	2	7	62	0,947	0,837	0,969	0,916
81-100	120	30	1	7	82	0,968	0,811	0,988	0,933
30-100	514	146	4	56	308	0,973	0,723	0,987	0,883

Tabela 5.5: Resultados da identificação de linguagem sem o conjunto de *tweets* avaliados como contendo várias línguas.

e filtrar o conjunto de *tweets* de forma a remover os *tweets* em inglês. Desta forma seria possível melhorar a abrangência.

5.2.3 Perda de *tweets*

Foram efectuados 2665 pedidos ao Twitter, desta forma foram obtidos 148417 *tweets*, sendo que 111646 destes se encontravam na BD. O TwitterEcho recolheu em média 75,2% dos *tweets* analisados. A média de recolha por utilizador foi de 80,2%.

Os resultados divididos por actividade média dos utilizadores encontram-se disponíveis na seguinte tabela:

Actividade média inferior	**Número de utilizadores**	***Tweets* na BD**	***Tweets* obtidos**	**Percentagem de *tweets* recolhidos**	**Percentagem de *tweets* recolhidos por utilizador**
1h	127	16073	21400	75.1	74.0
3h	217	26592	36463	72.9	72.8
6h	216	23153	32098	72.1	71.7
12h	277	19512	25287	77.2	76.3
1 dia	378	13078	16946	77.2	76.8
3 dias	725	9942	12197	81.5	83.2
6 dias	358	2046	2046	80.5	84.3
12 dias	228	841	999	84.2	87.3
24 dias	82	248	297	83.5	88.7
1 ano	57	161	189	85.2	90.4

Tabela 5.6: Resultados da recolha de *tweets* organizados por actividade média dos utilizadores.

O TwitterEcho efectua escalonamento de utilizadores (ver secção 3.3.4) verificando mais vezes utilizadores mais activos. Caso verificasse todos os utilizadores o mesmo número de vezes, a percentagem de *tweets* obtidos dos utilizadores muito activos seria extremamente baixa e dos poucos activos extremamente alta. Estes resultados permitem concluir que o escalonamento de utilizadores é de grande importância no TwitterEcho, permitindo equilibrar a percentagem de *tweets* recolhidos.

Como trabalho futuro é importante verificar os utilizadores extremamente activos, que produzem muitos *tweets* num curto espaço de tempo, com outros métodos da API do Twitter que possibilitem obter um maior número de *tweets* e não apenas o último.

Capítulo 6

Conclusões e Trabalho Futuro

O TwitterEcho permite a obtenção de dados em larga escala da comunidade portuguesa presente no Twitter. Fá-lo através do uso de um sistema distribuído, sendo este adaptável para outras comunidades. Devido ao uso de um sistema distribuído não se encontra tão dependente do Twitter e dos limites que este impõe na extracção de dados.

É modular, sendo por isso versátil, permitindo a adição incremental de novas funcionalidades. É robusto e fiável, encontrando-se em funcionamento há mais de 45 dias. Recolhe dados maioritariamente de utilizadores portugueses, sendo completamente autónomo no crescimento do conjunto de utilizadores a obter informações. Demonstrou bons resultados nos testes de avaliação, sendo que o módulo de análise de linguagem possui uma precisão excelente e abrangência razoável na identificação de português de Portugal.

Concluo que cumpri os objectivos e criei um sistema inovador e complexo. Este sistema é um contributo valioso, torna possível a realização de estudos profundos sobre a comunidade portuguesa presente no Twitter. Devido ao interesse deste sistema na presente data já foram desenvolvidos vários estudos com base nos dados recolhidos pelo TwitterEcho, ver secção 6.2.

6.1 Trabalho Futuro

Abordei o trabalho futuro ao longo da dissertação, nesta secção sintetizo o que referi anteriormente. O trabalho futuro pode ser divido em duas categorias principais: aumento de *performance* para tornar o sistema escalável para comunidades de grandes dimensões (por exemplo Brasil) e aumento da qualidade dos dados obtidos.

6.1.1 Aumento da *performance*

Os serviços podem sofrer alterações para serem executados menos vezes e mais rapidamente (ver secção 3.3.6). O módulo de análise de *tweets* (secção 3.4.1) pode ser melhorado para permitir a análise de mais de 170 mil *tweets* por dia. O módulo de adição de seguidores (secção 3.4.2) pode-se tornar um serviço para aumentar o número de seguidores adicionados por dia. A recolha de *tweets* (secção 3.4.4) pode ser realizada num sistema híbrido quando a Search API não devolve *tweets* suficientes a tarefa pode ser delegada para os clientes recorrendo à REST API.

6.1.2 Qualidade dos dados obtidos

Os clientes podem ser melhorados de forma a usarem outros métodos da REST API no caso de clientes muito activos, de forma a reduzir a perda de *tweets* (secção 5.2.3). As localidades e nomes disponíveis na BD para análise de perfil podem ser aumentadas de forma a melhorar a classificação por análise de perfil (secção 3.4.3). O módulo de identificação de linguagem pode ser estendido de forma a detectar a presença de múltiplas línguas e agir em conformidade (secção 5.2.2). Pode ser adicionado um módulo que seja um complemento da análise de perfil e linguagem, analisando as relações dos utilizadores para decidir se este é português (secção 5.2.1).

6.2 Aplicações

Os dados recolhidos pelo TwitterEcho são de grande interesse para vários estudos. Na presente data foram realizados os seguintes trabalhos tendo como base dados recolhidos por este:

- Twitómetro[1]: sistema que aferiu os sentimentos dos portugueses para as eleições de 5 de Junho de 2011. Teve grande impacto sendo mencionado em vários órgãos da comunicação social, como por exemplo no canal de televisão RTP [2].

- Identificação de comunidades: estudo realizado pela professora Eduarda Mendes Rodrigues, identificando três sub-comunidades. Esta identificação foi efectuada tendo em conta a interacção entre utilizadores.

- Identificação de *bots*: Submissão e aceitação de artigo para a conferência EPIA 2011 por Gustavo Laboreiro. Os dados usados neste artigo de identificação de *bots* foram recolhidos pelo TwitterEcho.

[1] http://legislativas.sapo.pt/2011/twitometro/
[2] http://www.youtube.com/results?search_query=twit%C3%B3metro&aq=f

- Twitter e TV: relacionei o número de referências no Twitter de programas televisivos e as audiências destes (apêndice B).

O SAPO Labs pretende manter o TwitterEcho em funcionamento e continuar a explorar os dados recolhidos por este. Na secção 1.3 de motivação referi estudos que podem ser replicados para a comunidade portuguesa, demonstrei o potencial deste sistema. Esse potencial já não faz parte apenas do futuro, já se concretizou, nomeadamente com o Twitómetro.

Referências

[ACT04] Bashir Ahmed, Sung-Hyuk Cha e Charles Tapper. Language identification from text using n-gram based cumulative frequency addition. In *Proc. of CSIS Research Day*, page 12.1–12.8, Pace University, NY, 2004.

[AMW10] Norhidayah Azman, David E. Millard e Mark J. Weal. Issues in measuring power and influence in the blogosphere. In *Web Science Conference 2010*, 2010.

[Arc] The Mail Archive. Re: [twitter-dev] whitelist status update. Disponível em http://dev.twitter.com/pages/streaming_api_concepts#authentication, acedido a última vez em 17 de Junho de 2011.

[BCM11] Peter Baker, Helene Cooper e Mark Mazzetti. Bin laden is dead, obama says, Maio 2011. Disponível em http://www.nytimes.com/2011/05/02/world/asia/osama-bin-laden-is-killed.html, acedido a última vez em 17 de Junho de 2011.

[Bea09] Claudine Beaumont. New york plane crash: Twitter breaks the news, again, Janeiro 2009. Disponível em http://www.telegraph.co.uk/technology/twitter/4269765/New-York-plane-crashTwitter-breaks-the-news-again.html, acedido a última vez em 17 de Junho de 2011.

[Blo08] Twitter Blog. Finding a perfect match, Julho 2008. Disponível em http://blog.twitter.com/2008/07/finding-perfect-match.html, acedido a última vez em 17 de Junho de 2011.

[BMRA10] Fabrício Benevenuto, Gabriel Magno, Tiago Rodrigues e Virgílio Almeida. Detecting spammers on twitter. *CEAS 2010 Seventh annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference*, Julho 2010.

[Car11] Kathryn Blaze Carlson. Tunisia's revolution: 'twitter saved my life', Janeiro 2011. Disponível em http://news.nationalpost.com/2011/01/21/tunisias-revolution-twitter-saved-my-life/, acedido a última vez em 17 de Junho de 2011.

[CHBG] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto e Krishna P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[Cho] Admin's Choice. Crontab – quick reference. Disponível em http://adminschoice.com/crontab-quick-reference, acedido a última vez em 17 de Junho de 2011.

[Cow] John Cowan. English trigram frequency table. Disponível em http://home.ccil.org/~cowan/trigrams, acedido a última vez em 17 de Junho de 2011.

[CT94] William B. Cavnar e John M. Trenkle. N-Gram-Based Text Categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US, 1994.

[dC] Portal do Cidadão. Atribuição do nome a um recém-nascido. Disponível em http://www.portaldocidadao.pt/PORTAL/entidades/MJ/IRN/pt/SER_atribuicao+do+nome+a+um+recem+nascido.htm, acedido a última vez em 17 de Junho de 2011.

[DF04] Daniel W. Drezner e Henry Farrell. The power and politics of blogs, 2004. Disponível em http://www.telegraph.co.uk/technology/twitter/4269765/New-York-plane-crash-Twitter-breaks-the-news-again.html, acedido a última vez em 17 de Junho de 2011.

[Dou] Nick Douglas. How levi's jeans duped the internet with their new secret ad. Disponível em http://gawker.com/388783/?tag=viralvideo, acedido a última vez em 17 de Junho de 2011.

[Dun94] Ted Dunning. Statistical Identification of Language. 1994.

[el] Apprendre en ligne. Analyse des fréquences en français. Disponível em http://www.apprendre-en-ligne.net/crypto/stat/francais.html, acedido a última vez em 17 de Junho de 2011.

[Elw99] David Elworthy. Language identification with confidence limits. *Computing Research Repository*, 1999.

[Fit] Will Fitzgerald. Language identi?cation a computational linguistics primer. Disponível em http://www.entish.org/lang-id-preso/lang-id-preso.pdf, acedido a última vez em 17 de Junho de 2011.

[fU] UNIXhelp for Users. Unix man pages: uptime. Disponível em http://unixhelp.ed.ac.uk/CGI/man-cgi?uptime, acedido a última vez em 17 de Junho de 2011.

[GAC+10] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic e Wolfgang Kellerer. Outtweeting the Twitterers — predicting information cascades in microblogs. In *Proceedings of the 3rd Workshop on Online Social Networks (WOSN 2010)*, June 2010.

[Gha11] Jeffrey Ghannam. In the middle east, this is not a facebook revolution, Fevereiro 2011. Disponível em http://www.washingtonpost.com/wp-dyn/content/article/2011/02/18/AR2011021806964.html, acedido a última vez em 17 de Junho de 2011.

[GNI] GNIP. Gnip twitter. Disponível em http://gnip.com/twitter, acedido a última vez em 17 de Junho de 2011.

[Gra] Tweet Grader. Top twitter cities. Disponível em http://twittergrader.com/top/cities, acedido a última vez em 17 de Junho de 2011.

[Gre95] G Grefenstette. Comparing two language identification schemes. In *3rd International Conference on the Statistical Analysis of Textual Data JADT'95*, 1995.

[hca] Twitter help center. O que é seguir? Disponível em http://support.twitter.com/articles/284901-o-que-xe9-seguir-novo-twitter, acedido a última vez em 17 de Junho de 2011.

[hcb] Twitter help center. O que é um # marcador? Disponível em http://support.twitter.com/articles/255508-o-que-xe9-um-marcador, acedido a última vez em 17 de Junho de 2011.

[hcc] Twitter help center. O que é um retweet? Disponível em http://support.twitter.com/articles/263102-o-que-xe9-um-retweet-novo-twitter, acedido a última vez em 17 de Junho de 2011.

[hcd] Twitter help center. O que são respostas e menções. Disponível em http://support.twitter.com/articles/252461-o-que-s-xe3-o-respostas-e-men-xe7-xf5-es-novo-twitter, acedido a última vez em 17 de Junho de 2011.

[HN99] Allan Heydon e Marc Najork. Mercator: A scalable, extensible web crawler. Technical report, Compaq Systems Research Center, Junho 1999.

[HRW08] Bernardo A. Huberman, Daniel M. Romero e Fang Wu. Social networks that matter: Twitter under the microscope. Dezembro 2008.

[JL10] Ryan Jansen e Jake Lussier. Cse 30332 programming paradigms final project report. Technical report, Department of Computer Science and Engineering University of Notre Dame, Maio 2010.

[JSFT07] Akshay Java, Xiaodan Song, Tim Finin e Belle Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop 2007*, Agosto 2007.

[JZSC09] B.J. Jansen, M. Zhang, K. Sobel e A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009.

[KB03] Ed Keller e Jon Berry. *The Influentials: One American in Ten Tells the Other Nine How to Vote, Where to Eat, and What to Buy*. The Free Press, 2003.

[KGA08] Balachander Krishnamurthy, Phillipa Gill e Martin Arlitt. A few chirps about twitter. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 19–24, New York, NY, USA, 2008. ACM.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park e Sue Moon. What is twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.

[Kra] Simon Kranig. Evaluation of language identification methods.

[LZT+09] L. F. Lopes, J. M. Zamite, B. C. Tavares, F. M. Couto, F. Silva e M. J. Silva. Automated social network epidemic data collector. *INForum informatics symposium*, 2009.

[Mar11] José Martins. Modeling user influence and expertise for news sources in social media. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2011.

[Med] Mediamonitor. Bem-vindo à mediamonitor. Disponível em http://www.mediamonitor.pt/, acedido a última vez em 17 de Junho de 2011.

[Mic] Microsoft. Windows live hotmail fact sheet sept. 30, 2010. Disponível em http://www.microsoft.com/presspass/presskits/windowslive/materials.aspx, acedido a última vez em 17 de Junho de 2011.

[Mou] Jorge Mourinha. A minha tv. Disponível em http://aminhatelevisao.blogspot.com, acedido a última vez em 17 de Junho de 2011.

[MR] Christopher Manning e Prabhakar Raghavan. Information retrieval and web search.

[MS] MPI-SWS. The twitter project page at mpi-sws. Disponível em http://twitter.mpi-sws.org/, acedido a última vez em 17 de Junho de 2011.

[oBC] MBC The Museum of Broadcast Communications. Share. Disponível em http://aminhatelevisao.blogspot.com, acedido a última vez em 17 de Junho de 2011.

[PHP] PHP. Image processing and gd. Disponível em http://php.net/manual/en/book.image.php, acedido a última vez em 17 de Junho de 2011.

[PP10] Alexander Pak e Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner e Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[Pra99] John M. Prager. Linguini: Language identification for multilingual documents. In *Journal of Management Information Systems*, pages 1–11, 1999.

[RGAH11] D. M. Romero, W. Galuba, S. Asur e B. A. Huberman. Influence and passivity in social media. *20th International World Wide Web Conference (WWW'11)*, 2011.

[Sch] Robin Schumacher. Using the new mysql query profiler. Disponível em http://dev.mysql.com/tech-resources/articles/using-new-query-profiler.html, acedido a última vez em 17 de Junho de 2011.

[SCHT07] Xiaodan Song, Yun Chi, Koji Hino e Belle L. Tseng. Information Flow Modeling based on Diffusion Rate for Prediction and Ranking. In *Proceedings of the WWW*, 2007.

[SIC] SIC. Peso pesado. Disponível em http://www.pesopesado.net/, acedido a última vez em 17 de Junho de 2011.

[Tel11] The Telegraph. Bin laden raid was revealed on twitter, Maio 2011. Disponível em http://www.telegraph.co.uk/technology/twitter/4269765/New-York-plane-crash-Twitter-breaks-the-news-again.html, acedido a última vez em 17 de Junho de 2011.

[TH] William J. Teahan e David J. Harper. Using compression based language models for text categorization.

[TMZ09a] TMZ. Michael jackson dies, Junho 2009. Disponível em http://www.tmz.com/2009/06/25/michael-jackson-dies-death-dead-cardiac-arrest/, acedido a última vez em 17 de Junho de 2011.

[TMZ09b] TMZ. Michael jackson memorial confirmed for staples center july 7, Julho 2009. Disponível em http://www.mtv.com/news/articles/1615268/michael-jackson-memorial-confirmed-july-7.jhtml, acedido a última vez em 17 de Junho de 2011.

[TVI] TVI. Perdidos na tribo. Disponível em http://www.tvi.iol.pt/perdidosnatribo/, acedido a última vez em 17 de Junho de 2011.

[Twia] Twitter. Authentication. Disponível em http://dev.twitter.com/pages/streaming_api_concepts#authentication, acedido a última vez em 17 de Junho de 2011.

[Twib] Twitter. Followers ids. Disponível em http://dev.twitter.com/doc/get/followers/ids, acedido a última vez em 17 de Junho de 2011.

[Twic] Twitter. Friends ids. Disponível em http://dev.twitter.com/doc/get/friends/ids, acedido a última vez em 17 de Junho de 2011.

[Twid] Twitter. Get search. Disponível em http://dev.twitter.com/doc/get/search, acedido a última vez em 17 de Junho de 2011.

[Twie] Twitter. Get statuses/public_timeline. Disponível em http://dev.twitter.com/doc/get/statuses/public_timeline, acedido a última vez em 17 de Junho de 2011.

[Twif] Twitter. Intro to developing for @twitterapi. Disponível em http://dev.twitter.com/pages/intro-to-twitterapi, acedido a última vez em 17 de Junho de 2011.

[Twig] Twitter. Rate limiting. Disponível em http://dev.twitter.com/pages/rate-limiting#search, acedido a última vez em 17 de Junho de 2011.

[Twih] Twitter. Rest api rate limiting. Disponível em http://dev.twitter.com/pages/rate-limiting#rest, acedido a última vez em 17 de Junho de 2011.

[Twii] Twitter. Statuses followers. Disponível em http://dev.twitter.com/doc/get/statuses/followers, acedido a última vez em 17 de Junho de 2011.

[Twij] Twitter. Statuses user_timeline. Disponível em http://dev.twitter.com/doc/get/statuses/user_timeline, acedido a última vez em 17 de Junho de 2011.

[Twik] Twitter. Streaming api: Methods. Disponível em http://dev.twitter.com/pages/streaming_api_methods#statuses-sample, acedido a última vez em 17 de Junho de 2011.

[Twil] Twitter. Streaming api: Methods. Disponível em http://dev.twitter.com/pages/streaming_api_methods#statuses-filter, acedido a última vez em 17 de Junho de 2011.

[Twim] Twitter. The twitter api. Disponível em http://dev.twitter.com/pages/api_overview, acedido a última vez em 17 de Junho de 2011.

[Twin] Twitter. Users lookup. Disponível em http://dev.twitter.com/doc/get/users/lookup, acedido a última vez em 17 de Junho de 2011.

[W3C] W3C. About w3c. Disponível em http://www.w3.org/Consortium/, acedido a última vez em 17 de Junho de 2011.

[Wan10] Alex Wang. Don't follow me: Spam detection in twitter. Julho 2010.

[We] Word-english. The 500 most commonly used words in the english language. Disponível em http://www.world-english.org/english500.htm, acedido a última vez em 17 de Junho de 2011.

[Wil] Dr. Ralph F. Wilson. The six simple principles of viral marketing. Disponível em http://www.wilsonweb.com/wmt5/viral-principles-clean.htm, acedido a última vez em 17 de Junho de 2011.

[Wil10] Fred Wilson. A look back at summize, Abril 2010. Disponível em http://www.businessinsider.com/a-look-back-at-summize-2010-4, acedido a última vez em 17 de Junho de 2011.

[Wor08] Jenna Wortham. Levi strauss scores viral gold with back-flipping jeans clip, Maio 2008. Disponível em http://www.wired.com/underwire/2008/05/levis-jeans-beh/, acedido a última vez em 17 de Junho de 2011.

[XLD11] XLDB. Reaction - xldb, Junho 2011. Disponível em http://xldb.fc.ul.pt/wiki/Reaction, acedido a última vez em 17 de Junho de 2011.

[Yah] Yahoo. Yahoo! maps web services - geocoding api. Disponível em http://developer.yahoo.com/maps/rest/V1/geocode.html, acedido a última vez em 17 de Junho de 2011.

[You11] YouTube. Susan boyle - britains got talent 2009 episode 1 - saturday 11th april | hd high quality, Abril 2011. Disponível em http://www.youtube.com/watch?v=RxPZh4AnWyk, acedido a última vez em 17 de Junho de 2011.

REFERÊNCIAS

Anexo A

Twitter API

O Twitter disponibiliza, através de uma interface *Web*, acesso a todas as informações dos utilizadores com perfil público. É também possível obter dados acedendo às páginas do Twitter dos utilizadores de quem se pretende informações. Aceder às páginas do Twitter possui desvantagens face à interface *Web*: tempo de resposta superior, obter dados não é directo (é necessário extrair do código HTML) e menor quantidade de informação (por exemplo os identificadores numéricos dos *tweets* não estão presentes na página do utilizador).

O Twitter possui três APIs distintas que são abordadas em detalhe nas próximas secções.

A.1 REST API

Esta interface é muito abrangente, permite efectuar praticamente todas as acções disponíveis no Twitter, publicar, seguir um utilizador, visualizar informações e *tweets* de um utilizador, entre outros.

Esta API possui diversos métodos com diferentes propósitos. De seguida abordo apenas os que são referidos na presente dissertação:

A.1.1 *Statuses/public_timeline*

Método que devolve vinte *tweets* recentes, sendo que armazena os resultados por 60 segundos, só devolvendo novos resultados após este período [Twie].

A.1.2 *Statuses/user_timeline*

Método que devolve *tweets* mais recentes de um utilizador. Este método numa única chamada permite devolver até duzentos *tweets* de um utilizador. Possui também paginação sendo possível devolver até 3200 *tweets* [Twij].

A.1.3 *Users/lookup*

Método que devolve informação do perfil de até cem utilizadores. No caso de o utilizador não possuir conta protegida devolve também o último *tweet*. Dados referentes a utilizadores cuja conta foi apagada ou suspensa não são enviados [Twin].

A.1.4 *Friends/ids*

Método que devolve os identificadores numéricos dos amigos de um utilizador [Twic].

A.1.5 *Followers/ids*

Método que devolve os identificadores numéricos dos seguidores de um utilizador [Twib].

A.1.6 *Statuses/followers*

Método que devolve informação do perfil de cem seguidores mais recentes de um utilizador. Os utilizadores que não possuem conta protegida devolve também o último *tweet* [Twii].

A.2 Search API

Esta interface tem como propósito a pesquisa de informação no Twitter. Permite a pesquisa de *tweets* que contenham determinadas palavras ou que sejam de um dado utilizador, entre outras pesquisas. Não se encontra integrada na REST API devido a motivos históricos e de compatibilidade [Twim]. Esta interface possui algumas características importantes: devolve identificadores numéricos dos utilizadores diferentes dos utilizadores REST API [Twid], posiciona os *tweets* por relevância, não envia os que não considera relevantes [Twif] e devolve no máximo 1500 *tweets* por chamada [Twid].

A.3 Sreaming API

A Streaming API permite obter *tweets* em quase tempo real do Twitter filtrando estes por palavras, utilizadores, entre outros parâmetros. Não permite obter dados anteriores ao momento em que a ligação é estabelecida.

Esta API possui diversos métodos com diferentes propósitos. De seguida abordo apenas os que são referidos na presente dissertação:

A.3.1 *Statuses/sample*

Este método devolve cerca de 1% de todos os *tweets* públicos. Acedendo a este[1] pode ser verificado que o Twitter usualmente devolve mais de 1000 *tweets* por minuto [Twik].

A.3.2 *Statuses/filter*

Este método devolve *tweets* que correspondem a determinados parâmetros. Permite pesquisar por palavras, *tweets* de um utilizador, etc. O nível de acesso por defeito permite o uso de 400 palavras e até 5000 utilizadores [Twil].

[1] http://stream.twitter.com/1/statuses/sample.json

A.4 Restrições de utilização

A REST API encontra-se limitada no número de pedidos. São permitidos 150 pedidos anónimos por hora (este limite é imposto em função do IP) e 350 pedidos autenticados (*OAuth*) por hora (este limite é imposto em função da conta) [Twih]. É possível efectuar um pedido de aumento do número de pedidos permitidos (*whitelist*). Os critérios de aceitação não estão definidos publicamente. Existem casos de *whitelists* concedidas que foram retiradas sem qualquer justificação. É usual não atribuir *whitelist* a projectos de investigação universitária [Arc].

A Search API não necessita de autenticação, e tal como a REST API limita o número de pedidos. Este limite é imposto em função do IP, não é público mas é superior ao da REST API [Twig]. Testei os limites até cerca de vinte mil pedidos por hora a partir do mesmo IP, sem ter sido bloqueado.

O uso da Streaming API obriga ao uso de autenticação (básica ou *OAuth*) [Twia] e encontra-se limitada a uma ligação por conta. Muitos dos métodos disponíveis nesta API apenas estão disponíveis para utilizadores a quem o Twitter atribuiu acesso privilegiado.

Anexo B

Twitter e TV

No presente apêndice relaciono o número de referências no Twitter com as audiências atingidas por programas televisivos, no mês de Maio. Os programas são: Peso Pesado [SIC] e Perdidos na Tribo [TVI].

Quantificar o número de referências de cada programa requer descobrir que palavras são usadas no Twitter para se referirem ao programa. No caso do Peso Pesado, espera-se que a forma mais usual será pelo nome do programa, mas podem existir outras, fui usando sucessivas pesquisas e inspecção manual para descobrir novas formas:

1. select text from status where text like '%peso%pesado%' and text not like '%peso pesado%';
2. select text from status where text like '%peso%pesado%' and text not like '%#pesopesado%' and text not like '%peso pesado%';

Descobri #pesospesados, PesoPesado, pesosPesados, pesos pesados, peso-pesado, #peso_pesado, gordos SIC. A instrução SQL final que abrange todos os casos é:

select text from status where (text like '%peso%' and text like '%pesado%') or (text like '%gordos%' and text like '% sic %');

Procedi da mesma forma para Perdidos na Tribo e cheguei à seguinte instrução SQL:

select text from status where (text like '%perdidos%' and text like '%tribo%') or (text like '%tribo%' and text like '% tvi %');

Procedi à avaliação da precisão destas pesquisas, obtive para ambas as instruções SQL 200 *tweets* aleatórios. No caso da instrução de pesquisa para o programa "Peso Pesado" 198 *tweets* referem-se ao programa, precisão de 99%. No caso da instrução de pesquisa para o programa "Perdidos na Tribo" 199 referem-se ao programa, possui portanto uma precisão de 99,5%. Em ambos os casos a precisão é elevada e similar, é importante ser similar para não existirem enviusamento das amostras e favorecimento.

Usei audiências disponibilizadas pela MediaMonitor do grupo MarkTest [Med] [Mou], disponibilizam dois indicadores: *share* e *rating*. *Share* quantifica a percentagem de pessoas que está a visualizar um programa e se encontra a ver televisão, o *rating* não tem em conta se a pessoa se encontra a ver televisão. Optei por usar o share porque reflecte melhor a competição independente do número de pessoas a ver televisão em cada dia [oBC].

B.1 Peso Pesado

O Peso Pesado estreou dia 1 de Maio na SIC e consiste num conjunto de pessoas que pretendem emagrecer e se encontram numa quinta sobre orientação de treinadores e nutricionistas.

É exibido em formato de diários de 40min, usualmente ao domingo existem episódios especiais de duração superior, cerca de uma hora e meia, nos quais é efectuado um balanço da semana e é expulso alguém. Espera-se que este episódio desperte mais interesse que os diários. O Peso Pesado não foi transmitido nos dias 5, 14, 18 e 29 de Maio. As três primeiras datas devem-se à transmissão de jogos de futebol e a última aos Globos de Ouro.

Os *tweets* referentes ao Peso Pesado foram agrupados por datas:

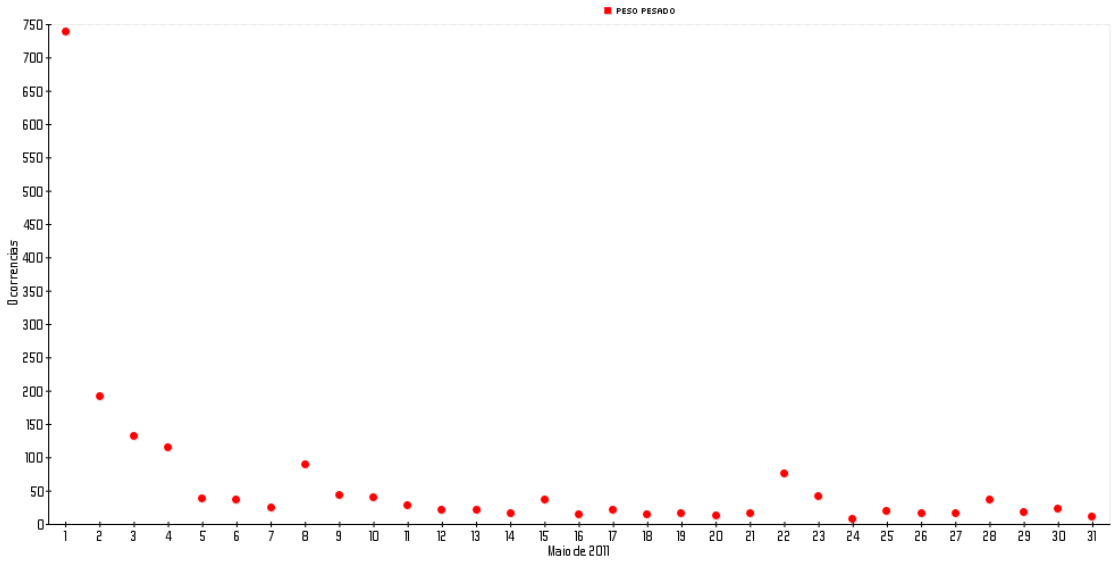


Figura B.1: Número de *tweets* mencionando Peso Pesado no mês de Maio.

É possível verificar:

- No dia de estreia 1 de Maio teve grande impacto no Twitter, é um indicador forte de audiências elevadas nesse dia.
- Nos primeiros três dias após a estreia continua a ser bastante mencionado no Twitter, isso deve-se possivelmente a ser novidade e as pessoas tentarem decidir se vale a pena visualizar, as audiências devem permanecer altas.
- No dia 5 de Maio existe uma queda abrupta do número de referências no Twitter, neste dia não foi transmitido o Peso Pesado.
- No dia 8 de Maio o número de referências aumenta, é o episódio especial ao domingo, neste dia estreou o Perdidos na Tribo na TVI no mesmo horário em concorrência directa.
- É possível identificar apenas olhando para a figura X dois episódios especiais devido a surgirem com um número de referências bastante superior à vizinhança.

- Os episódios especiais de dia 15 e dia 28 produziram pouco impacto no Twitter, é um indício de poucas audiências quando comparando com os outros episódios especiais a 1, 8 e 22 de Maio.

O *share* do Peso Pesado relativo ao mesmo período de tempo encontra-se na seguinte figura:

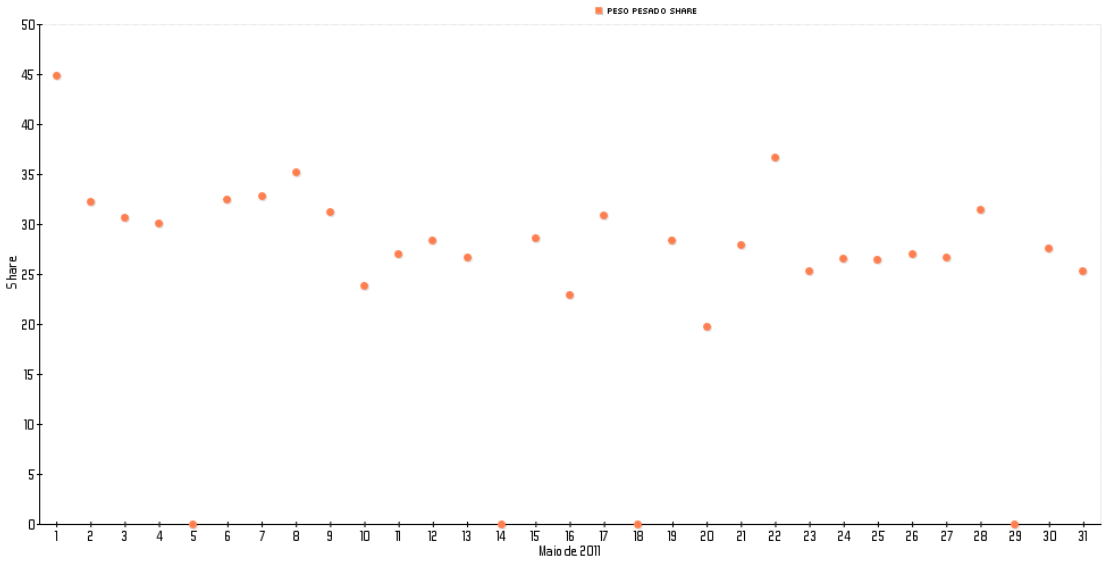


Figura B.2: *Share* do Peso Pesado no mês de Maio.

É possível verificar:

- A estreia foi um sucesso.
- Nos primeiros três dias após a estreia continua a possuir audiências elevadas comparando com outros diários tais como dia 11 e 12 de Maio ou 23 a 27 de Maio.
- É possível identificar com facilidade os dias de expulsão pois possuem audiências superiores a vizinhança.
- Os episódios especiais de dia 15 e dia 28 produziram menor audiência quando comparando com os outros episódios especiais a 1, 8 e 22 de Maio.

Em ambos os casos é possível retirar conclusões similares, na figura B.3 é possível verificar as similaridades (os episódios especiais encontram-se assinalados).

B.2 Perdidos na Tribo

O Perdidos na Tribo estreou-se no dia 8 de Maio na TVI e trata-se de um programa televisivo em que três grupos compostos por pessoas conhecidas são enviadas para três tribos diferentes, tendo que viver com estas e se adaptarem aos seus costumes e forma de viver arcaica.

A TVI optou por outro formato exibindo apenas episódios ao domingo, em concorrência directa com os episódios especiais do Peso Pesado.

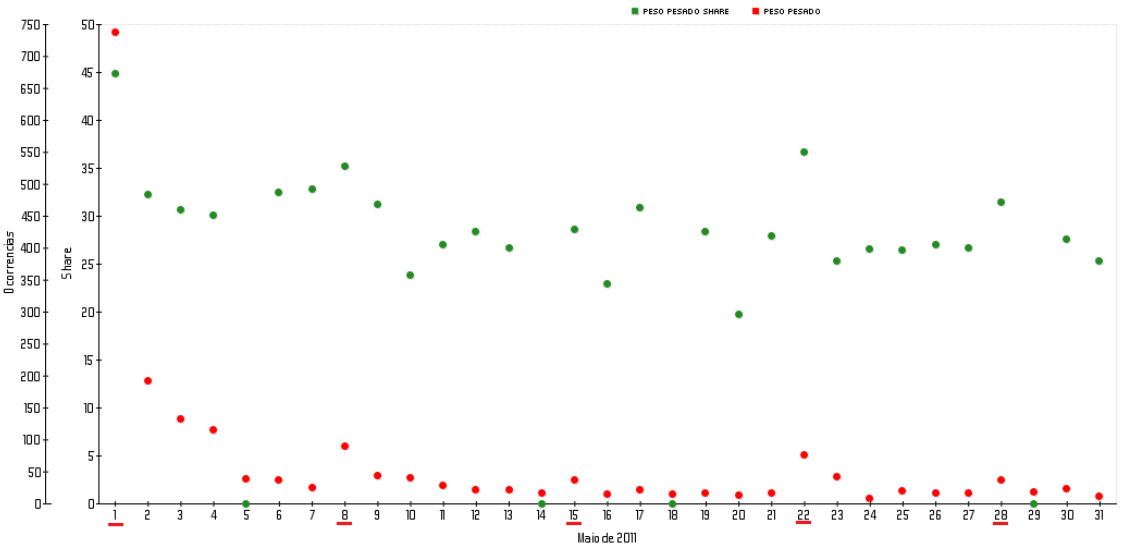


Figura B.3: Número de *tweets* e *share* do Peso Pesado no mês de Maio.

Os *tweets* referentes ao Perdidos na Tribo foram agrupados por datas tendo em conta apenas o mês de Maio, figura B.4.

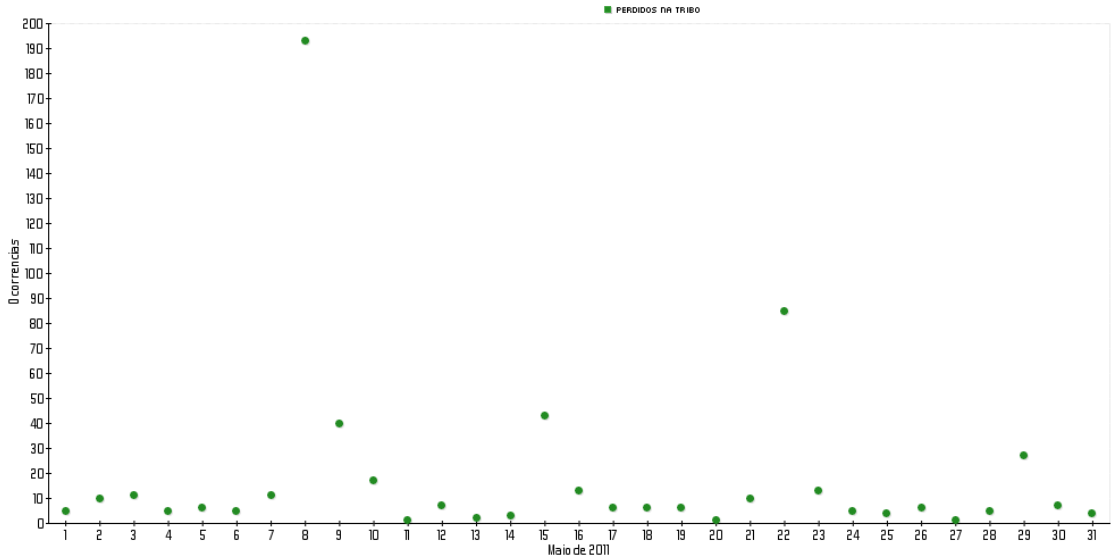


Figura B.4: Número de *tweets* mencionando Perdidos na Tribo no mês de Maio.

É possível verificar:

- No dia de estreia 8 de Maio teve grande impacto no Twitter tanto nesse dia como no dia a seguir, é um indicador forte que teve audiências elevadas.
- O impacto da estreia foi tão grande que os utilizadores do Twitter continuaram a falar no programa no dia seguinte.

- É possível identificar facilmente os dias de transmissão 8, 15, 22 e 29 devido a diferença significativa do número de referências face a vizinhança.
- O episódio com menor número de referências foi no dia 29 isto é um indício que foi o episódio com menores audiências.

O *share* do Perdidos na Tribo relativo ao mesmo período de tempo encontra-se na figura B.5.

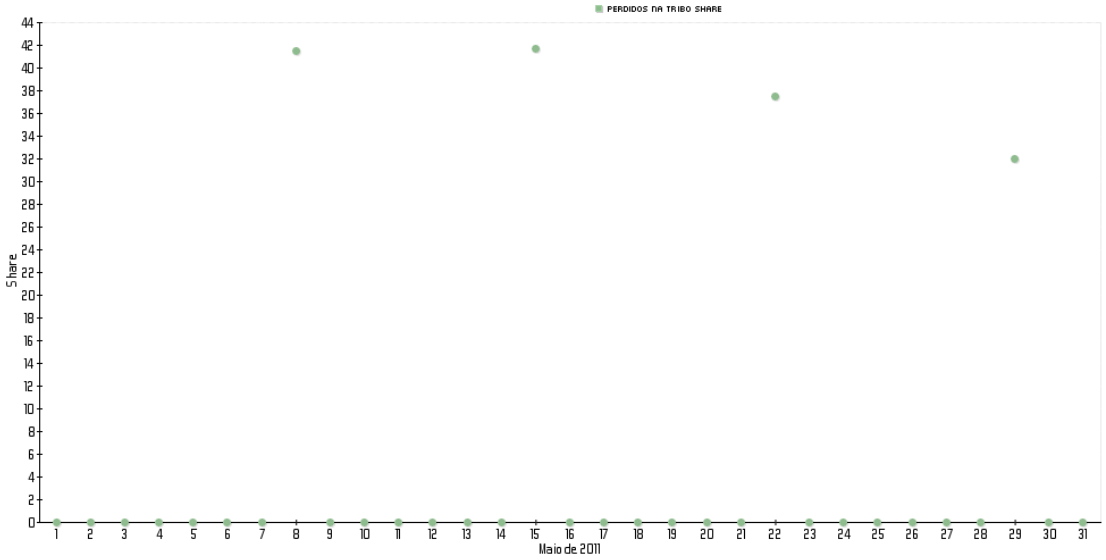


Figura B.5: *Share* do Perdidos na Tribo no mês de Maio.

Em ambos os casos é possível retirar conclusões similares, na figura B.6 é possível verificar as similaridades entre o número de referências e o *share*.

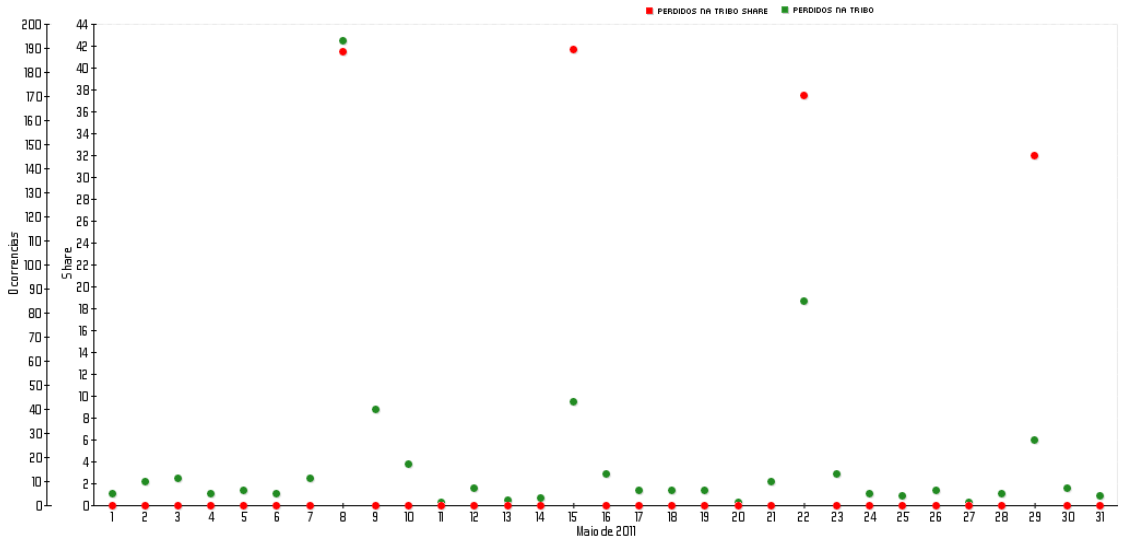


Figura B.6: Número de *tweets* e *share* do Perdidos na Tribo no mês de Maio.

Em ambos os programas (Peso Pesado e Perdidos na Tribo) é possível verificar que existe uma correlação entre o número de referências no Twitter e o *share* televisivo.

B.3 Peso Pesado vs Perdidos na Tribo

É possível efectuar uma comparação do número de referências de ambos os programas:

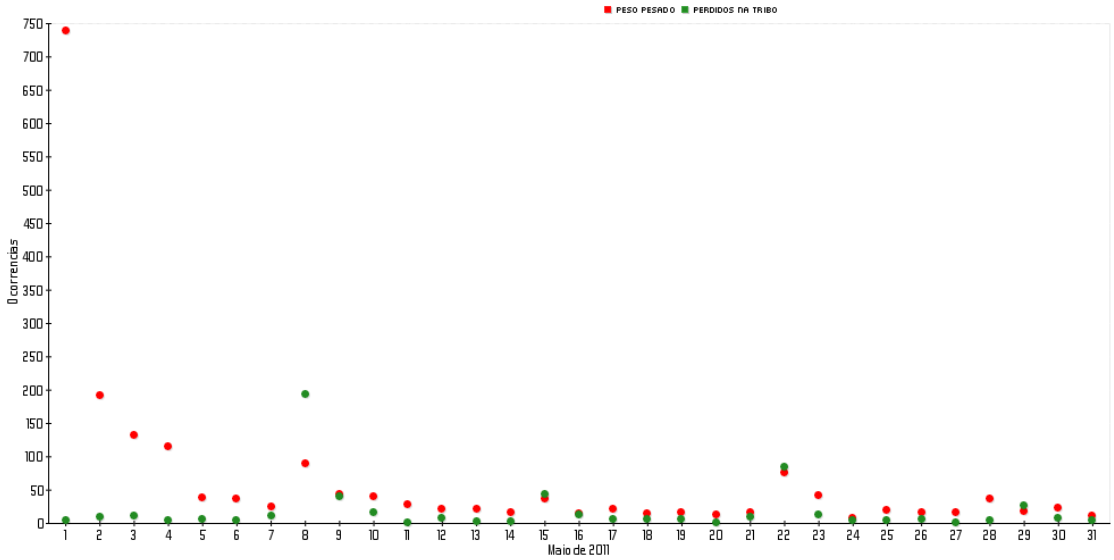


Figura B.7: Número de *tweets* mencionando Peso Pesado e Perdidos na Tribo no mês de Maio.

Esta análise possui indícios fortes que o Perdidos na Tribo na sua estreia dia 8 de Maio teve maiores audiências que o Peso Pesado, no entanto o Peso Pesado na sua estreia dia 1 de Maio parece ter tido maior impacto que o Perdidos na Tribo na sua estreia. Nos dia 15 e 22 as diferenças não são significativas o suficiente para retirar conclusões. No dia 28 não concorreram e no dia 29 o Perdidos na Tribo concorreu com os Globos de Ouros.

O *share* de ambos os programas encontram-se representados na figura B.8.

Como é possível verificar as ilações retiradas anteriormente através de análise do número de referências confirmam-se. O Perdidos na Tribo ganhou na sua estreia, no entanto o Peso Pesado teve uma estreia com maiores audiências que a estreia do Perdidos na Tribo. As audiências mais baixas do Perdidos na Tribo coincidem com a noite dos Globos de Ouro que teve cerca de 47,2% de share face aos 32,0% do Perdidos na Tribo.

Concluo que existe uma correlação forte entre referências no Twitter e audiências. Este estudo pode ser melhorando, além do número de referências, pode ser tido em conta: número de utilizadores distintos que referiram, análise de sentimentos, entre outros.

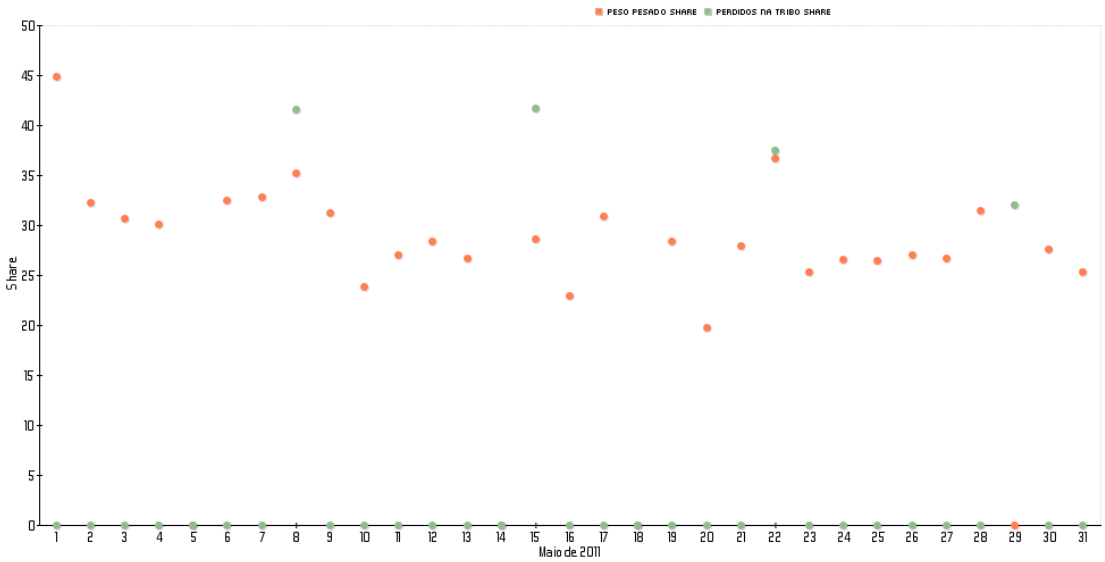


Figura B.8: *Share* do Peso Pesado e Perdidos na Tribo no mês de Maio.

Anexo C

Medidas de Desempenho

No contexto de tarefas de classificação é usual o uso das medidas de desempenho: precisão, abrangência, taxa de negativos verdadeiros e exactidão.

Por exemplo ao classificar um documento como estando escrito em português, as seguintes categorias podem ocorrer:

- TP, *true positive*: o documento foi classificado correctamente como estando escrito em português.
- FP, *false positive*: o documento foi classificado como estando escrito em português e não se encontra escrito em português.
- FN, *false negative*: o documento foi classificado como não estando escrito em português e encontra-se escrito em português.
- TN, *true negative*: o documento foi classificando correctamente como não estando escrito em português.

Precisão (P) é definida como o número total de documentos classificados correctamente como estando escrito em português sobre o número total de classificações positivas:

$$P = \frac{TP}{TP + FP} \tag{C.1}$$

Uma precisão de 1 significa que todos os documentos classificados como estando escritos em português, foram classificados correctamente (FP=0). Esta medida não possui informação sobre a quantidade de documentos escritos em português classificados.

Abrangência (A) é definida como o número total de documentos classificados correctamente como estando escrito em português sobre o total de documentos escritos em português:

$$A = \frac{TP}{TP + FN} \tag{C.2}$$

Uma abrangência de 1 significa que todos os documentos escritos em português foram classificados correctamente (FN=0). Esta medida não possui informação sobre a quantidade de documentos que não se encontram escritos em português e foram classificados erradamente como tal.

É comum existir uma relação inversa entre precisão e abrangência, usualmente é possível aumentar um reduzindo o outro.

Taxa de negativos verdadeiros (TNV) é definida como o número de documentos classificados correctamente como não estando escritos em português sobre o número de documentos que não se encontram escritos em português:

$$TNV = \frac{TN}{TN + FP} \tag{C.3}$$

Exactidão (E) é definida como o número de classificações correctas sobre todas as classificações:

$$E = \frac{TP + TN}{TP + FP + FN + TN} \tag{C.4}$$