Memoria

VII Congreso Internacional de Computación y Telecomunicaciones

del 23 al 26 de setiembre de 2015

Lima - Perú

Universidad
Inca Garcilaso de la Vega
Nuevos Tiempos. Nuevas Ideas
Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

Detecção de Eventos Sociais com Dados do Twitter

Augusto Zangrandi, Luis A. Rivera, Ausberto Castro, Fermin Tang

zangrandii@gmail.com, {rivera, ascv, tang}@uenf.br

Laborátorio de Ciências Matemáticas – LCMAT
Universidade Estadual do Norte Fluminense – UENF
Av. Alberto Lamego, 2000; CEP 28015-620, Campos dos Goytacazes – RJ – Brasil

Resumo*: Detecção de eventos consiste no processo de identificação de padrões de mudança relevantes em um sistema. O modelo facilita a detecção de algum evento social prévio análise dos dados criados por usuários no serviço de rede social Twitter, serviço que tem recebido grande atenção acadêmica por sua alta popularidade e característica de troca de informações em tempo real. O modelo utiliza, como exemplo de evento, manifestações que ocorrem no território brasileiro durante o mês de agosto de 2014. As publicações são obtidas de Twitter através da sua interface para desenvolvedores, são convertidas para o seu modelo de espaço vetorial representativo que são classificação com máquina de vetores de suporte. A classificação consiste na divisão das publicações obtidas em duas classes: positivas e negativas. Sendo positivas as que dizem respeito a uma manifestação real, com local e horário, e negativas as que somente mencionam manifestação, mas não dizem respeito ao acontecimento real. Após classificadas, as publicações são agrupadas e exibidas em gráficos no formato de série-temporal e mapas de marcadores, aonde é possível detectar, através da análise visual, tanto o horário dos eventos, através de picos que surgem nos gráficos, quanto a sua localização, através de sua concentração em determinadas regiões do mapa.*

Palavras Chave*: Twitter, Evento Social, Detecção de Evento, Análise de Redes Sociais.*

Abstract*: Event detection consists in the identification of relevant patterns of change in a system. The model implements event detection with data from the publications created by the users in the social networking service Twitter, one service that has received great scholarly attention due to its high popularity and characteristic of real time information exchange. The model uses, as an example of event, manifestations that occur in Brazilian territory, during the month of August 2014. First, the model obtain the publications through the Twitter interface for developers. Then, the publications are converted to their vector space model, for extracting their feature vectors so that they can be classified by the support vector machine. The classification consists on the division of the publications into two groups: positives and negatives. The positives means that the publication referes to a real manifestation, with time and location, and the negatives means that the publication only mentions the word, but not a real occurrence. Once classified, the publications are grouped and displayed in time-series graphs and marker maps, where it's possible to detect, by visual analyse, both the time of the event, through peaks that appear in the graph, and their location, through aglomeration in certain regions of the map.*

Keywords*: Twitter, Social Event, Event Detection, Social Network Analysis.*

1. Introdução

A **detecção de eventos** é o processo de identificação de acontecimentos que fogem das regras normais de funcionamento de um sistema computacional, ou de padrões de mudanças relevantes dentro dos mesmos. O *sistema de monitoramento de tópicos em documentos de texto* é um exemplo. Neste sistema, um evento pode ser o repentino surgimento de documentos contendo um termo específico ou um novo tópico, e a frequência maior ou menor deste evento pode ser fundamental para o sistema.

Como parte dos sistemas, existem os *sensores* que são os componentes que produzem diretamente as *entradas* para a detecção. As saídas dos sensores são os dados vindos diretamente da análise do ambiente - temperatura, velocidade, som, documentos de texto, e outras mídias - e servem como entrada para a detecção de eventos (Sakaki et al, 2010). Um *usuário* de um serviço de uma rede social (SNS: *Social Networking Services*) é considerado um sensor virtual ou sensor social, e cada mensagem é considerada como uma informação sensorial.

Os SNSs são as principais formas que as pessoas se relacionam no ambiente virtual. São criadas diariamente milhões de publicações sobre os mais variados temas, o que tem chamado atenção acadêmica crescente nas áreas de mineração de dados, a detecção de eventos. Entre os SNSs mais populares está o Twitter, com sua base de 500 milhões de usuários e publicações, onde os usuários produzem dinamicamente uma imensa quantidade de informações trocadas, em tempo real dos acontecimentos do mundo. O usuário, neste contexto, atua como um sensor dos eventos do mundo real, produzindo informações que podem ser utilizadas para detectá-los.

Para a detecção dos ventos sociais, existem vários métodos; sendo os principais os métodos estatísticos, métodos probabilísticos e métodos de aprendizado de máquina. Os **métodos estatísticos** analisam os dados vindos dos sensores e na criação de um modelo estatístico para eles. Através da comparação entre os dados reais e o modelo estatístico, é possível indicar se em determinado momento, ocorre um evento fora do padrão ou não, através da diferença entre os valores esperados e reais. Os **métodos probabilísticos** criam modelos probabilísticos para indicar, a partir dos dados estatísticos, qual a probabilidade de ocorrer um evento em determinado momento. Enquanto os métodos de **aprendizado de máquina** analisam os padrões existentes em um conjunto de dados vindos dos sensores para determinar o

funcionamento geral do sistema. São geralmente aplicados em dados esparsos e em sistemas que necessitam de alto desempenho computacional.

Neste trabalho se desenvolve um modelo de detecção de eventos através do Twitter, utilizando o modelo de aprendizado de máquina SVM (Support Vector Machine) e a ajuda de gráficos de série-temporal interativos para detectar erupções anômalas de publicações, o que, na prática, determina a ocorrência de um evento. O modelo utiliza como exemplo as manifestações ocorridas em Brasil no período de 01 a 31 de agosto de 2014.

O trabalho é organizado da seguinte forma: na Seção 2 abordam-se a detecção de eventos sociais. Na Seção 3 se formula o modelo de detecção de eventos através do Twitter; na Seção 4 aborda-se os detalhes da implementação do modelo e análise de resultados; na Seção 5, finalmente, conclui-se com a indicação de trabalhos futuros.

2. Detecção de eventos sociais

A **detecção de eventos** é o processo de identificação de um evento dentro de um sistema. Um **evento** é um padrão de mudança significativo ou ocorrência anômala em relação ao comportamento geral do sistema observado. A *detecção de eventos*, envolve ocorrências significativas detectadas dentro do sistema.

O sistema de detecção de eventos deve ser capaz de transformar os dados vindos dos sensores e identificar os eventos inerentes a esses dados. Os dados dos sensores são dados de baixo-nível, sendo medições diretas de uma característica do mundo, e a detecção deve transformá-lo em dados de alto nível de forma que seja possível a compreensão humana. Para realizar este fato, o método deve agregar, converter e reformatar os dados recebidos em uma estrutura independente da fonte de dados (Fienberge e Shmueli, 2005).

Jiang et al. (2009) categorizam em três classes de sistemas:

- *Natural e artificial*: Em sua classificação mais básica, sistemas podem ser de origem natural ou artificial. Sistemas naturais são aqueles já presentes na natureza, enquanto que os são criados pelo homem.
- *Observável e não-observável*: Sistemas observáveis são aqueles onde suas características podem ser observadas pelo homem, sem a necessidade de um sensor específico, como monitorar se está de dia ou de noite. Sistemas não-observáveis necessitam da implementação de um sensor específico, como monitorar se a temperatura de um ambiente está acima de 40ºC.
- *Qualitativo e quantitativo* (método de análise): Sistemas qualitativos são analisados de acordo com suas saídas diretamente. Sistemas quantitativos, são analisados de acordo com a medição de performance ou métricas derivadas das saídas do sistema.

No modelo implementado, os eventos são detectados em um sistema artificial, não-observável e analisados qualitativamente. Sistemas não-observáveis, por sua vez, necessitam de sensores, para quantificar os dados do ambiente e permitirem a detecção de eventos.

2.1. Sensores

Para a detecção de eventos dentro de sistemas não-observáveis, são utilizados sensores que quantificam e medem as informações presentes no ambiente em questão. Os sensores são quaisquer componentes produtores dos dados e quantificadores do ambiente analisado. Um canal de publicações de noticias pode ser um sensor, na medida que o sistema de detecção de eventos analise seus documentos de texto como formato de entrada de dados.

Um sensor pode ser, também, um usuário de um serviço de rede social, ao criar uma publicação sobre um determinado acontecimento, possibilitando a aplicação da detecção de evento de um determinado acontecimento dentro de uma gama de outras publicações. O sensor, neste âmbito, é denominado como sensor social.

Os recursos para a implementação da detecção de eventos são os dados gerados diretamente pelos sensores, neste caso, uma publicação ou através de algum dado indireto considerado relevante.

2.2. Eventos em documentos de texto

Os eventos são detectados a traves de análise padrões presentes nos documentos texto, referenciando com eventos do mundo real. Um evento, neste caso, indica uma ocorrência significativa no contexto de interesse de alguma atividade humana, tal como relacionados com shows musicais, festas, políticos, modas, e outros.

Weng e Lee (2011) classificam os métodos de detecção de eventos em documentos de texto em dois tipos: documento-pivô e recursivo-pivô. Os métodos de *documento-pivô* baseiam-se na divisão de documentos em grupos de acordo com a similaridade léxica de seus conteúdos, porém, como não existem regras para implementação da detecção, também não existe padrão para a criação dos algoritmos. São considerados alguns critérios técnicos: a) proximidade temporal, em que os documentos referentes ao mesmo evento costumam ser próximos temporalmente; b) erupção de documentos similares, em que o espaço de tempo entre a erupção de documentos similares geralmente indica eventos diferentes; c) mudanças de frequência, onde as mudanças rápidas nas frequências de um termo geralmente é sinal de documento referente a um novo evento. Os métodos do tipo *recursivo-pivô* analisam a distribuição e a associação das palavras. Também, não existe uma maneira melhor para se implementar, e cada caso deve ser analisado em sua unicidade, porém Sakaki et al. (2010) citam três recursos para a implementação: estatísticos (número de palavras e a posição da palavra-chave dentro do documento); palavras-chave (palavras de referência no documento); contexto de palavra (palavras antes e depois da palavra-chave). Sem embargo, Aiello et al. (2013) consideram que os dois tipos possuem desvantagens. Os métodos documento-pivô possuem problemas com fragmentação de grupos e, no contexto de aquisição de documentos em tempo real, eles dependem de limiares arbitrários para a inclusão de um documento em um grupo. Os métodos recursivo-pivô geralmente fazem associações errôneas entre palavras-chave.

2.3. Sensores sociais

Os *serviços das redes sociais* são plataformas online onde os usuários podem se relacionar criando perfis, compartilhando publicações de variados temas e acontecimentos, e atualizações em formato texto, foto, áudio e vídeo. As redes sociais mais populares, segundo a lista10.org[7], são: Facebook, Youtube, Qzone, Sina Weibo, WhatsApp, Google+, Tumblr, Line, Twitter, WeChat, entre outros; cada um com suas característica de serviços. O Facebook é o serviço com maior base de usuários, seguido por WhatsApp.

Os serviços disponíveis possuem interfaces para desenvolvedores de maneira que as informações sejam adquiridas e analisadas de forma sistêmica. Isto possibilita a criação de vários tipos de serviços externos que se conectam aos servidores das SNSs para se obter os dados gerados pelos usuários e analisar para os mais variados fins. Um dos serviços externos é a detecção de eventos que, segundo Dong et al. (2014), é um dos tópicos mais importantes na análise de redes sociais.

Relacionado às SNSs estão os *microblogs*, que são uma maneira de compartilhar as informações no formato texto curto, permitindo aos usuários fazerem rápidas atualizações. O mais popular nesta categoria é o Twitter. No entanto, o microblog é um conceito para outras ferramentas como Facebook e Google+, na forma de atualização de *status*, que é uma forma do usuário compartilhar a informação do que está passando.

Pelo tamanho reduzido das publicações nos microblogs, o esforço necessário para a geração de informação é menor, o que potencializa e adiciona dinamismo a distribuição de experiências entre usuários. Ao vivenciar um evento, o usuário pode sentir a necessidade de compartilhar com seu grupo de amigos e eles com outros. Por sua característica pessoal e de tempo real, eles se tornam uma fonte única de informação sobre todo os do tipo de acontecimentos do mundo real (Mai e Hranac, 2013).

O usuário, ao realizar uma publicação em seu microblog, passa a atuar como um sensor de acontecimentos do mundo real. Parecido como nos sensores físicos, o usuário ao vivenciar um acontecimento e publicar sobre ele, está agindo como um sensor social do mesmo. A Figura 1 ilustra um usuário como sensor de acontecimentos do mundo real e produtor de documentos que são armazenados no Twitter, criando publicações sobre eventos naturais, esportivos e desastres não-naturais.

Twitter

O Twitter, como serviço de microblog, permite compartilhar publicações de texto até 140 caracteres. A principal característica do serviço é o dinamismo das publicações e a sua facilidade de compartilhamento. O dinamismo do serviço é explicitado pela funcionalidade de *trending topics* (assuntos do momento), onde estão ranqueados os termos mais comentados do momento, com atualizações várias vezes por dia.

[1]http://lista10.org/tech-web/as-10-maiores-redes-sociais-do-mundo/

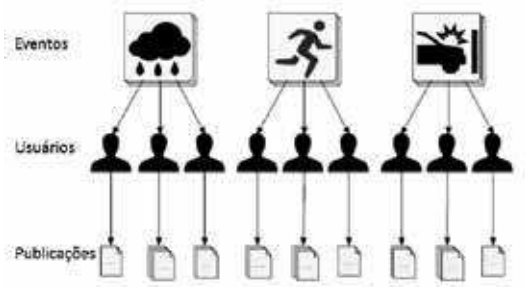

Figura 1: Usuários como sensores de eventos

O recurso *seguir* do Twitter permite que um usuário U1, ao seguir outro usuário U2, passa a receber todas as publicações de U2 na sua página de *linha do tempo*. A linha do tempo é a principal página de interação com o sistema, onde são exibidas todas as publicações dos usuários que se segue. Para cada publicação é possível curtir (gostar publicamente), responder ou retwittar (passar a mensagem para os usuários seguidores).

O trabalho de Java et al. (2007) apresenta a formação de comunidades e a motivação das pessoas ao utilizarem os serviços de microblogs do Twitter, e concluem que essa forma de interação induz a alta reciprocidade e correlação, indicando um entendimento mútuo entre os usuários e a facilidade de pulverização de informação. Jansen et al. (2009) estudaram o uso do Twitter como uma ferramenta de transferência de informação de pessoa para pessoa e recomendam que o Twitter é uma das peças chave para o monitoramento de informação. Matuszka et al. (2013) estudam o Twitter e os assuntos do momento, e uma forma de sumarizá-los e de retenção de informações e histórias, para que elas não se percam devido à intensa criação de informações proporcionada pelo serviço.

2.4. Trabalhos relacionados

Entre os trabalhos de detecção de eventos aplicados nos mais variados âmbitos, alguns aplicaram métodos probabilísticos para categorizar as medições em grupos de interesse, e outros utilizaram métodos de aprendizado de máquina para treinar e categorizar futuras ocorrências. Na maioria desses trabalhos utilizaram-se técnicas no âmbito do Twitter. Tem-se a Gupchup et al. (2009) que utilizam a técnica de Análise de Componente Principal (ACP) para construir um modelo capaz de coletar as tendências das medidas de uma rede de sensores sem fio para detectar anomalias. Hong et al. (2014) utilizam a detecção de eventos para detectar intrusos em redes de computadores. Abou-Zleikha et al. (2014) desenvolvem um algoritmo de decisão de floresta aleatória (*random forest*) para detectar eventos em dados vocais, onde os eventos em questão podem ser momentos de silêncio, risadas, e outros. No trabalho de Ihler et al. (2006) é construído um modelo de Poisson variável no tempo para detectar eventos anômalos em dados de contagem de séries temporais.

Sakaki et al. (2010) desenvolveram uma técnica que detecta terremotos e tufões no Japão através das publicações coletadas do Twitter. As palavras "terremoto" e "tremendo" são utilizadas como palavras-chave, e foi utilizado o método de aprendizado "máquina de vetores de suporte" com o kernel linear. Takahashi et al. (2011) desenvolvem um sistema que monitora publicações e

detecta ocorrências de rinite alérgica no Japão. A aplicação monitora publicações que contêm a expressão "hayfever" (rinite alérgica) enviadas ao Twitter. Ao analisar a correlação entre os dados obtidos pelos sensores sociais e pelos sensores já utilizados, Takahashi et al chegaram a conclusão de que o Twitter pode ser utilizado, neste caso, como uma alternativa aos sensores já existentes. Vinceller e Laki (2013) analisam o ciclo de vida de cada palavra chave comum de detectar eventos, e enfatizam que o aparecimento de eventos específicos em redes sociais podem surgir antecipadamente aos outros meios de comunicação.

Mai e Hranac (2013) analisaram se as publicações enviadas ao Twitter pode ser uma fonte de dados para acidentes de transporte. O trabalho utiliza a API de transmissão em tempo real do Twitter para detectar publicações relevantes contendo palavras-chave como "acidente", "batida", "rodovia" e apenas as publicações contendo a localização geográfica do usuário foram selecionadas. Porém um sofisticado filtro de conteúdo e localização se mostrou necessário para maximizar a relevância dos dados. Wang et al. (2013) desenvolveram um algoritmo para a detecção de palavras que erupcionam repentinamente no Twitter. As palavras são obtidas através da interface para desenvolvedores "Streaming API". É utilizado o pré-processamento de palavras para remover replicações como risadas. É utilizado um modelo probabilístico de mistura gaussiana para a extração das palavras que erupcionam e para a detecção de evento. Finalmente, para o reconhecimento da localização é utilizado um modelo probabilístico de campo aleatório condicional (CRF: *conditional random field*).

3. Detecção de ventos através do Twitter

O modelo de detector de eventos proposto neste trabalho utiliza as publicações disponibilizadas pelo Twitter como fonte de dados para detectar ocorrências de manifestações públicas. O detector se apoia nas informações criadas por seus usuários para retirar informações relevantes como o local e o horário de manifestações. Para validar o detector de eventos, o modelo busca por publicações que contém uma palavra-chave. Todas as publicações Twitter, contendo a palavra-chave, são copiadas para um ambiente de trabalho local. Mesmo contendo a palavra-chave buscada, muitas podem não dizer respeito ao evento que o modelo deseja detectar. Para classificar as publicações que são relevantes para o modelo e as que devem ser descartadas, o modelo utiliza a técnica de aprendizado de máquina SVM (*Support Vector Machine*) como classificador de texto. O arquivo local, separado em publicações de treino e publicações de teste, logo de serem caracterizado, é calculado os parâmetros da máquina de aprendizado por SVM utilizando-se publicações de treino. A essa fase denomina-se representação do conhecimento. Com os elementos de teste é realizado a detecção de eventos, ao se realizar operações matemáticas que determinam as relações de distância entre os dados, podendo inferir a qual classificação cada dado se enquadra. Na Figura 2 se ilustra a operação descrita.

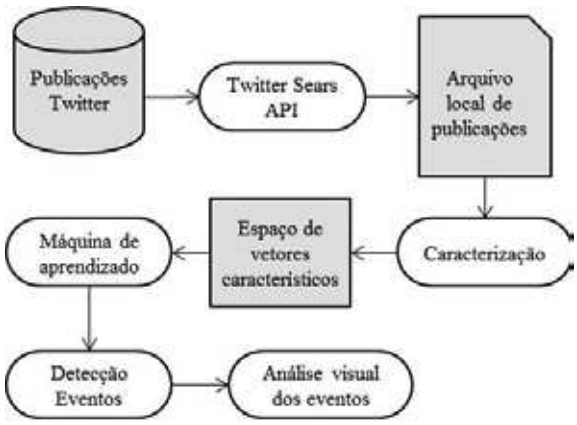

Figura 2: Processo de detecção de eventos

Após a classificação das publicações, o modelo extrai o horário e a localização das publicações positivas. A informação do horário será utilizada para agrupar as publicações, como intuito de determinar os horários em que ocorreram mais ocorrências de menções às manifestações. Os horários de pico representam os eventos, que são ocorrências anômalas ao funcionamento normal do fluxo contínuo de publicações. Já a localização será utilizada para a análise visual das regiões de ocorrência dos eventos, que serão traçadas nos mapas apresentados.

3.1. Publicações como fonte de dados

No Twitter são criadas diariamente cerca de 500 milhões de publicações[8], e junto com elas, diversos dados úteis são gravados. Cada publicação textusl possui também dados adicionais como: horário de criação, idioma, dados do usuário (nome, localização, descrição e foto) que publicou, geolocalização (para envios de smartphones, tablets e notebooks com opção ativada), referências a links, usuários e hashtags.

As publicações, por serem criadas por usuários, não passam por algum tipo de filtro, e grande parte delas devem ser descartadas para o sucesso das análises, seja por conter palavras com a grafia errada, gírias, ou por não dizerem respeito a qualquer conteúdo relevante. De acordo com Kelly (2009), aproximadamente 40% das publicações não possuem relevância, atuando apenas como ruído para as análises.

Um evento será considerado como tal se houver uma quantidade suficiente de publicações relacionadas. Para Sakaki et al. (2010), os fatores que possibilitam a existência de eventos são: escala, influência e região. O *fator escala* se refere às vivencias de muitas pessoas que geram maior quantidade de publicações; a *influência* é relacionada ao impacto do evento na vida das pessoas para compartilhar experiências; enquanto a *região* é o espaço e tempo, que permite realização de estimativas e localização de eventos. Eventos com grande escala, com alto grau de influência e com região de tempo e espaço, são os mais propícios para a aplicação da detecção de eventos, pois geram maior fluxo de publicações e permitem a estimativa de sua localização. Exemplos de

[2] www.internetlivestats.com/twiter-statistics/

eventos com essas características são desastres naturais como terremotos, tempestades, ciclones e tufões; eventos sociais como grandes festivais, eventos esportivos e políticos; e desastres não naturais como os acidentes.

3.2. Interfaces para obtenção dos dados

Para disponibilizar as publicações, o serviço utiliza a forma de notação JSON (Java Script Object Notation) para troca de dados. Uma formatação leve, em formato de texto, que permite que os dados das publicações sejam transferidos pela rede de internet para serviços externos. Os dados são estruturados em pares de nome/valor e possuem ordem fixa. O Código 1 ilustra a estrutura de uma publicação, com dados de criação, id, texto, etc.

Código 1: Estrutura JSON de uma publicação

```
1  {
2  :created_at=> "ThuJul 31 15:14:27 +0000 2014",
3  :id=> 494863667100258304,
4  :text=> "Manifestacao deixa transito
   congestionado em São Cristovao: Funcionarios
   do transporte alternativo protestam...",
5  (...)
6  :user=>{
7  :id=>2340427167,
8  :name=> "Rodrigo",
9  :location=> "Sao Paulo",
10 (...)
11 },
12 :geo=>nil,
13 :coordinates=>nil,
14 :place=>nil,
15 (...),
16 :lang=> "pt"
17 }
```

O Twitter disponibiliza interfaces de APIs (SearchAPI, REST API e Streaming API) para permitir acesso às informações disponíveis em seu serviço. Search API permite buscar por publicações por palavras chave e devolve a quantidade relevante das publicações. Das informações obtidas são selecionados os dados de interesse: texto da publicação, hora de criação, latitude e longitude da posição do usuário, localização configurada do perfil do usuário e identificador único da publicação.

Os dados selecionados são gravados localmente em um arquivo CSV (*comma separated values*), um formato de dados em texto em que os campos são separados por vírgulas. Os dados formam uma tabela, sendo as vírgulas delimitadoras das colunas e as quebra de linhas delimitadoras de linhas. São adicionadas aspas nos campos para impedir que as vírgulas dos textos se confundam com as vírgulas separadoras dos campos. Os dados que interessa para o propósito deste trabalho são: Id, data e hora, texto, latitude, longitude, e localização.

3.3. Caracterização

Os dados dos arquivos CSV são publicações textuais que devem ser convertidos em entidades numéricas para suas respectivas operações matemáticas. Por se tratar de informações de tamanhos variados e compostos de palavras e expressões, essas são convertidas as entidades numéricas vetoriais. Cada vetor representa uma publicação, e o conjunto selecionado de publicações passa a ser um conjunto de vetores definindo o espaço vetorial.

Os vetores são constituídos por termos de índice, que podem conter pesos de acordo com sua importância ou não (Salton et al, 1975). É usado o conceito de recursos, que são termos separados por espaços, frases separadas por ponto e vírgula, e quebra de linhas. Para Joachims (1998), os recursos provenientes de documentos de texto possuem as características: *alta dimensionalidade* (cada palavra é um termo, e a publicação é grande); *baixa irrelevância* (poucos termos irrelevantes, mantendo alta dimensionalidade); *densidade* (vetores com poucos elementos diferentes de zero).

O recurso escolhido para este modelo é a presença ou ausência de palavras no *dicionário de termos*. Uma *palavra* é um termo constituído de cadeia de caracteres limitada por espaços. Na tokenização, cada publicação deve ser tokenizada de forma a se obter os respectivos termos constituintes. Estes termos passam por pré-processamento, que agrupa termos com mesmo significado, considerando palavras seguidas de pontuação e maiúsculas ou minúsculas. Todos os termos são agrupados e ordenados em ordem alfabética, criando assim o dicionário de termos. No dicionário, cada termo possui um único identificador definido por sua ordem. A dimensão do vetor característico é definida pelo número de palavras no dicionário. Um vetor característico de uma publicação inicialmente é zero, indicando que todos os termos são ausentes, que passa ser presente (valor 1) a posição do vetor indicado pelo índice da posição do termo no dicionário. A sequência de caracterização descrita aqui é ilustrada pela Figura 3.

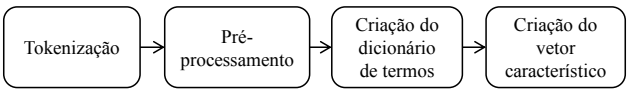

Figura 3: Processo de conversão de caracterização.

Tokenização

Sequência de palavras da publicação é preparada em termos básicos. Para Turney e Pantel (2010), alguns tokenizadores devem ser capazes de reconhecer termos de mais de uma palavra, como "Lula da Silva", termos contendo hífen e pontuações, ignorando pronomes e preposições, entre outros. Neste trabalho se estabelece espaços como limitadores de termos, e utilizar uma lista de palavras recorrentes, como preposições, para ignorá-las na criação de termos, como "em", "no", "de", e outros. Os links começando com "http://" ou "www." também são ignorados. Exemplo, uma publicação contendo a frase "manifestação acontecendo no Rio" será tokenizado em três termos: "manifestação", "acontecendo", "Rio".

Pré-processamento

Nessa etapa os termos são corrigidos para mais simples. As palavras com letras maiúsculas devem passar todo a minúsculas, removidos os caracteres especiais e as pontuações, de forma que os termos sejam iguais que são essencialmente iguais. Ou seja, é retirada a diferenciação dos termos como, por exemplo, "manifestação", "Manifestação", "manifestação!" e "manifestação." Isso diminui a complexidade da análise dos dados, na medida em que reduz a dimensão dos vetores característicos.

Vetor característico

Para cada publicação é criado um vetor da dimensão do dicionário, inicialmente com valores nulos. Posteriormente, para cada término da publicação é consultado no dicionário o índice do termo. Esse índice indica a posição desse termo no vetor que deve ser registrado com o dígito 1 indicando a presencia do termo.

Considere-se, por exemplo, um dicionário contendo cinco termos como a sequência de pares de índice e termo: (1,"acontecendo"); (2,"campos"); (3,"gosto"); (4,"manifestação"); (5,"não"). Considere duas mensagens A=("manifestação", "acontecendo", "campos") e B = ("não", "gosto", "manifestação"). A publicação A gera um vetor Va = (1, 1, 0, 1, 0) porque os termos de A estão no dicionário com índices 4, 1 e 2. Enquanto a publicação B gera um vetor Vb = (0, 0, 1, 1, 1) porque os termos de B ocupam índice 5, 3, e 4 do dicionário indicando presencia, nessas posições, no vetor.

3.4. Extração de dados

A extração e o agrupamento dos dados da publicação, como horário e localização, são relevantes para a sua demonstração no ambiente interativo. Para extrair o horário de um evento, o modelo utiliza apenas informação contida na estrutura da publicação. Para extrair a localização, são utilizadas três fontes de dados, na ordem: a geolocalização, a cidade na definição do perfil do usuário e a cidade no texto da publicação. A geolocalização indica a localização, latitude e longitude, do evento gerado. Este dado é obtido através dos dispositivos GPS presentes nos smarphones, tablets e alguns notebooks.

A geolocalização, porém, só está presente em um pequeno número de publicações, seja porque o dispositivo não tem GPS ou não está ativado. Para as publicações sem GPS, a localização pode ser extraída do perfil do usuário. No perfil, o dado de localização pode conter nome da cidade ou não. Caso exista a cidade, será consultada a lista de cidades do país, neste caso brasileira. No caso que não conste cidade alguma, deve-se verificar se o texto da publicação possui alguma cidade válida.

3.5. Máquina de aprendizado

O método SVM permite construir uma máquina que permite classificar as publicações entre relevantes ou não para a análise de eventos. Essa máquina é definida em base de um conjunto de informações, chamados dados de treino, calculando os parâmetros do modelo segundo os procedimentos estabelecidos no método supervisionado SVM. Este método é adequado para classificações de texto por possuir proteção contra o ajuste demasiado ao conjunto de dados analisados e utilizar funções de kernel, que são capazes de lidar com os dados de alta dimensão sem perda de performance (Joachims, 1998).

O método SVM foi desenvolvido por Vapnik nos anos 90, para tratar problemas de classificação de elementos distribuídos no espaço d-dimensional, enfocados como problemas de otimização. Neste casso, os elementos do espaço d-dimensional são vetores característicos que devem ser classificados em duas classes: relevantes ou não. Para isso, o SVM tenta achar um plano de dimensão n, que seria um hiperplano, que separa os vetores com a maior margem possível. O algoritmo parte do princípio que existe uma diferença fundamental que separa as duas classes e cria a separação espacial entre elas. O método deduz que, após a criação do hiperplano, os vetores de um lado do hiperplano fazem parte de uma classe e os do outro lado de outra classe, o que na prática se mostra funcionar (Grigorik, 2008).

O problema de otimização é formulado como, dado um conjunto de treinamento de n vetores característicos d-dimensionais x_i e suas respectivas classes y_i que o vetor pertence,

$$D = \{(x_i, y_i) | x_i \in R^d, y_i \in \{-1,1\}\}_{i=1,\dots,n},$$

O objetivo é encontrar o hiperplano P que divide os pontos que possuem $y_i = 1$ dos que possuem $y_i = -1$. O hiperplano P da forma $w^T x + b = 0$, sendo w o vetor normal do plano hiperplano, e b o deslocamento em relação à origem do sistema. É sabido esse hiperplano é limitado por dois hiperplanos de suporte $w^T x + b = 1$ e $w^T x + b = -1$, por y_i possuir valor 1 ou -1, como ilustra a Figura 4.

Na forma reduzida, o modelo é reformulado como:

$$y_i(w^T x_i + b) \geq 1, \text{ para } 1 \leq i \leq n.$$

Somente os pontos mais próximos, chamados de vetores de suporte, influenciam a criação dos hiperplanos. A distância entre os hiperplanos de suporte, definido pela projeção mínima da diferença entre os vetores de suporte positivo x_+ e negativo x_- sobre vetor w, é a margem $\frac{2}{\|w\|} = \frac{w}{\|w\|}(x_+ - x_-)$. O problema de otimização do SVM é maximizar a margem, ou equivale a minimizar $\|w\|$, formulado como:

Minimizar $\|w\|$

Sujeito a $y_i(w^T x_i + b) \geq 1$, para $1 \leq i \leq n$.

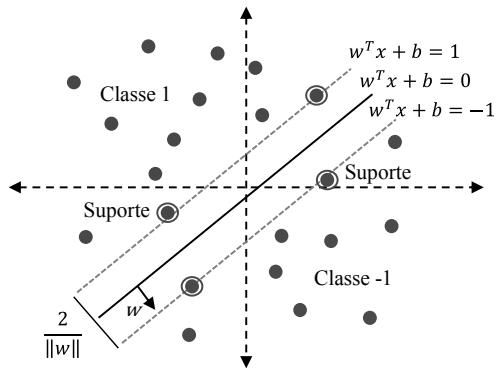

Figura 4: Planos no limiar dos pontos.

Podem existir casos em que os dados não sejam linearmente separáveis, porém existe uma margem muito pequena entre os vetores de suporte, que para margem um pouco maior deva se ignorado certos pontos como sendo de uma ou outra classe. Esse caso é considerado com SVM de margem suave, formulado como:

Minimizar $\|w\| + C \sum \varepsilon_i$ (1)

Sujeito a $y_i(w^T x_i + b) \geq 1 - \varepsilon_i$, para $1 \leq i \leq n$.

Mesmo assim, o SVM pode ter dificuldade de classificação linear dos dados, então são utilizadas as funções kernel que transformam os dados para que sejam mais facilmente classificáveis, tal como utilizados por Ali e Smith (2005), Ozer et al (2011), e Megri e Naqa (2014).

Resolver essa formulação suave, neste caso, utilizando os dados de treino, passa a ser o treinamento onde vão ser calculados os parâmetros w e b, tendo como entrada $\{(x_i, y_i)\}_{i=1,\dots,n}$ e C.

A verificação é feita com os dados de teste, ingressando unicamente os vetores característicos x_i ao modelo e esperando obter y_i desejado:

$$w^T x_i + b = y_i$$

Uma vez confirmado, possivelmente com uma margem de aproximação aceitável, o modelo será utilizado para a classificação dos eventos no formato vetor característico.

4. Implementação e análise dos resultados

O modelo é implementado através de classes nativas da linguagem Ruby[9] e de interfaces de comunicação com outros serviços como a Search API do Twitter e a biblioteca LIBSVM.

As publicações foram obtidas por Search API do Twitter pela palavra chave "manifestação", entre 01 e 31 de agosto de 2014. Foram realizadas várias buscas em diferentes datas, devido às políticas de Twitter na liberação das publicações limitada por uma semana de antiguidade de criação. As publicações são salvas em um arquivo CSV, e separadas em arquivos de treino e teste. Para manipular os arquivos é utilizada a biblioteca CSV nativa do Ruby.

Nas publicações para fase de treino são adicionados os valores de classe "1" para positivos e "-1" para negativos. Por exemplo, no formato (id, horário, texto, latitude, longitude, localização-perfil, classe), são colocados dois casos na Tabela 1.

Tabela 1: Exemplo de duas publicações

Dados	Positiva	Negativa
Id	497444091421286400	497439186807685121
Hora	2014-08-07 14:45:23	2014-08-07 14:40:19
Texto	Aumento da passagem de ônibus provoca manifestação em Sorocaba	O sorriso é a manifestação dos lábios
Lat.	-15.850 902	-21.167 484
Long.	-47.944 792	-41.331 175
Loc-perfil	Brasil	Rio de Janeiro
Classe	1	-1

A conversão para o modelo de espaço vetorial é feita utilizando classes para manipulação de cadeias de caracteres (String) e vetores (Array) do Ruby. Através de métodos dessas classes é possível aplicar todos os passos da conversão de tokenização, pré-processamento, criação do dicionário de termos e vetores de características.

4.1. Treino e teste

A classificação das publicações é realizada no nível dos vetores característicos com o uso de SVM facilitada pela biblioteca LIBSVM. A interface para Ruby permite que a biblioteca seja utilizada através de classes da linguagem e que a classificação seja aplicada de forma integrada com o resto do processo. Assim, a interface "rb-libsvm"[10] de LIBSVM, na versão 1.1.5, permite treinar, testar e usar um modelo de SVM. Para treinar, utilizam-se as funções *Libsvm::Problem* e *Libsvm::Parameter*. A função *Libsvm::Problem* é encargada de receber o conjunto de n descritores de treino $\{(x_i, y_i)\}_{i=1,\dots,n}$, que são os vetores característicos x_i e suas respectivas classes y_i. Enquanto a função *Libsvm::Parameter* encapsula os ajustes dos parâmetros do SVM. Para o modelo desejado, apenas o parâmetro de custo C da expressão (1) é ajustado, porém também é necessário inicializar os valores dos parâmetros ε_i (eps) e cache_size, como:

```
@problema = Libsvm::Problem.new
@parametro = Libsvm::SvmParameter.new
@parametro.cache_sezi = 1
@parametro.eps = 0.001
@parametro.c = 0.1  #cost
```

Para o treino, as publicações são lidas a partir do arquivo CSV como a seguir:

```
def carregador_publicacoes_treino caminho
 CSV.open(caminho) do |csv|
  csv.each do |linha|
    @publicacoes_treino<<Publicacao.new(linha,self)
  end
 end
end
```

As classes y_i da publicação está na posição 6 de cada linha do arquivo CSV, portanto basta iterar entre as publicações e guardar essa posição de cada uma. A inserção dos descritores e os parâmetros no LIBSVM, e treino do classificador são realizados pelos métodos *Libsvm::problem.set_examples* e *Libsvm::Model.train*. O primeiro recebe apenas os descritores e os une na estrutura da interna da biblioteca, o segundo recebe esses dados em conjunto com os parâmetros e treina buscando a solução para a expressão (1), e obtendo os valores de w e b, tal como ilustrada pela Figura 5.

Para saber se o modelo possui uma boa taxa de acertos ou não, deve ser testado com os respectivos valores de teste, utilizando o método *Libsvm::Problem.predict*. Casso a taxa de acertos for menor, deve ser treinar novamente variando os valores iniciais de C, tal como ilustrado pela Tabela 2, onde $C = 0.1$ deu um acerto de 90.5% com um performance de aceitável em relação aos valores.

4.2. Resultados

Após a classificação, é feita a extração dos dados do horário e localização. O horário é extraído diretamente a partir da informação presente no arquivo CSV. A localização é extraída de acordo explicado em 3.4.

Para a criação do ambiente foi utilizada a ferramenta para desenvolvimento de aplicações web Ruby on Rails4. No ambiente são exibidos gráficos de série-temporal das

[3] http://ruby-lang.org

[4] https://github.com/febeling/rb-libsvm

publicações, divididas primeiramente por dias, e posteriormente por horas. É possível visualizar cada faixa de horário da publicação em um mapa de marcadores.

Tabela 2: Taxa de acerto SVM.

C	Taxa de acerto	Performance
…	50%	…
0.001	50%	9.4s
0.01	78.5%	9.3s
0.1	90.5%	8.0s
1	89.5%	7.9s
10	86.5%	7.6s
100	86.5%	7.8s
1000	86.5%	7.5s
…	86.5%	…

Os dados utilizados forma 13.611 publicações positivas e 16.587 negativas. As publicações positivas e negativas exibem divergências no que diz respeito à extração das cidades e geolocalizações. Desses dados, se observa que 60% dos positivos foram possíveis obter as localizações e apenas um 1.3% tiveram as geolocalizações (ver Tabela 3). Isso demonstra que a grande maioria das publicações ainda está sendo enviada pelo computador ou por dispositivos sem o GPS ativado.

Tabela 3: Classificação de dados extraídos.

Publicações	Quantidade	Com Cidade	Com GPS
Positivas	13.611	8.148 (60%)	177 (1,3%)
Negativas	16.587	8.700 (52,4%)	610 (3,6%)
Total	30.198	16.848 (56%)	787 (1,9%)

Análise Visual

Para analisar os eventos, no ambiente interativo (disponível em: http://deteccao.zangrandi.me/) são exibidos gráficos de séries-temporais das publicações, criados a partir da ferramenta Highcharts[11] e mapas de marcadores criados a partir de TileMill[12]. Nesses gráficos é possível enxergar a existência de picos de publicações em determinados horários, o que podem indicar eventos, e de onde estão surgindo essas publicações.

Na Figura 5 é mostrada a frequência de publicações relacionadas às manifestações durante cada dia do mês de agosto. Observa-se que existe uma maior concentração de 1484 publicações no dia 20, e nos dias 23 e 24 mostra uma baixa significativa, possivelmente corresponde ao sábado e domingo respectivamente, período que em Brasil houve convocatórias dos últimos protestos sociais.

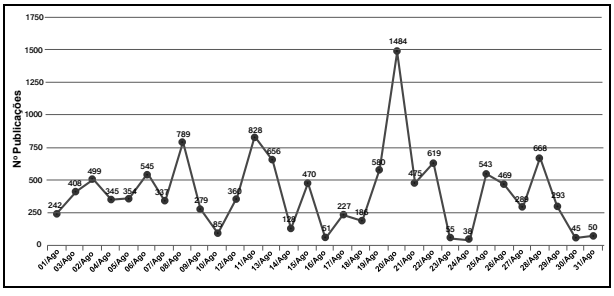

Figura 5: Publicações do mês de agosto de 2014

No dia 20 poderá se ver os eventos ao longo das horas nesse dia. A Figura 6 ilustra o comportamento dos eventos de protesta, se enfatizando entre as 7 a 10 de manhã, tentando de subir no final da tarde, sinal que no início do dia foi a convocatória.

Ao clicar no ponto entre as 6h e 7h no dia 20 de agosto, são exibidos os eventos distribuídos no território brasileiro, tal como ilustra a Figura 7. As publicações que não contém a geolocalização são aproximadas através do mapeamento das cidades para a sua geolocalização, de acordo com a informação do IBGE[13].

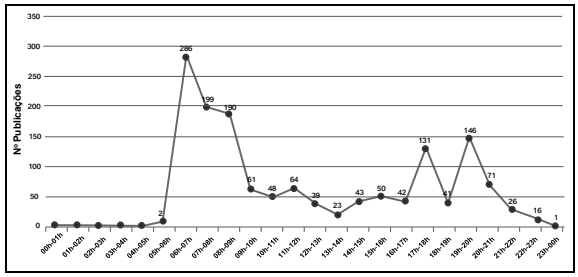

Figura 6: Publicações do dia 20 de agosto

Na imagem é possível visualizar uma maior concentração de publicações vindas de determinada área do estado de São Paulo. Ao aplicar zoom no mapa, o agrupamento de publicações se desfaz, permitindo que as publicações sejam visualizadas de modo agrupado com mais exatidão, ou também de modo único.

Figura 7: Mapa de marcadores para horário entre 6h a 7h

5. Conclusões e trabalhos futuros

Junto com outros trabalhos que obtiveram sucesso ao detectar outros eventos com dados do Twitter, como o de Sakaki et al. (2010) que detecta terremotos, e Mai et al. (2013) que detecta acidentes de trânsito, o modelo implementado apresentou que é possível a detecção de manifestações através dos dados obtidos do serviço. Manifestação é um acontecimento com certa escala, de importância na vida das pessoas e que possui local e horário, o que possibilita a sua detecção, pois, os usuários ao tomarem conhecimento de uma manifestação, rapidamente publicam sobre ela, gerando picos de publicações que podem ser observados através dos gráficos de série-temporal.

A classificação com SVM utilizou um conjunto de dados de treino do SVM de cerca de 1% dos dados totais, e

[5] http://www.highcharts.com
[6] https://www.mapbox.com/tilemill/
[7] http://ibge.gov.br

obteve cerca de 90% de taxa de acerto, produzindo resultado relevante e permitindo que, a partir de uma pequena gama de amostras, o conjunto completo dos dados fosse classificado com baixa margem de erro.

O modelo implementado ainda não consegue estimar de forma automática se, em um determinado momento no tempo, está ocorrendo um evento ou não, ou seja, se a quantidade de publicações em determinada faixa de horário é normal ao funcionamento do sistema ou de fato anômala. Isso seria possível através da implementação de um modelo probabilístico, que se encarregaria de analisar a qual tipo de distribuição probabilística os dados se enquadram. Seria feita então a comparação entre a distribuição e os dados reais obtidos.

Referências bibliográficas

Abou-Zleikaha, M. et al. (2014) Non-linguistic vocal event detection using online random forest. In: *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, *37th International Convention on.* [S.l.: s.n.] p. 1326-1330.

Aiello, L. M. et al. (2013) Sensing trending topics in twitter. IEEE.

Ali, S.; Smith, K. A. (2005) Kernal width selection for svm classification: A meta-learning approach. *International Journal of Data Warehousing and Mining*, IGI Publishing, v.1.

Dong, X. et al. (2014) Multiscale event detection in social media. *CoRR*, abs/1404.7048.

Fienberg, S. E.; Shmueli, G. (2005) Statistical issues and challenges associated with rapid detection of bio-terrorist attacks. John Wiley and Sons.

Gao, D. et al. (2014) Sequential summarization: A full view of twitter trending topics. *Audio, Speech, and Language Processing*, IEEE/ACM Transactions on, v.22, n. 2, p. 293-302, Fevereiro 2014. ISSN 2329-9290.

Grigorik, I. (2008) *Support Vetor Machines (SVM) in Ruby.* Em:

https://www.igvita.com/2008/01/07/support-vector-machines-svm-in-ruby/. Acesso em: 30/06/2014.

Gupchup, J. et al. (2009). Model-based event detection in wireless sensor networks. *CoRR* abs/0901.3923.

Hong, J.; Liu, C. C.; Govindarasu, M. (2014) Detection of cyber intrusions using network-based multicast messages for substation automation. In: *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES.* [S.l.: s.n.], p. 1-5.

Ihler, A.; Hutchins, J.; Smyth,P. (2006) Adaptive event detection with time-varying poisson processes. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: ACM (KDD'06) p. 207-216. ISBN 1-59593-339-5.

Jansen, B.J. et al. (2009) Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, ASIS and T.

Java, A. et al. (2007) Why we twitter: Understanding microblogging usage and communities. *9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, ACM.

Jiang, W.; Blumberg, A. F.; Buttrey, S. E. (2009) Event detection challenges, methods, and applications in natural and artificial systems. *14th ICCRTS: "C2 and Agility"*, Lockheed Martin MS2.

Joachims, T. (1998) Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98,* Springer Berlin Heidelberg, p. 137-142.

Kelly, R. (2009) Twitter Study Reveals Interesting Results About Usage – 40is Pointless Babble. Disponível em:

http://www.pearanalytics.com/blog/2009/twiter-study-reveals-interesting-results-40-percent-pointless-babble/. Acesso em: 25/03/2014.

Mai, E.; Hranac, R. (2013) Twitter interactions as data source for transportation incidents. *TRB 2013 Annual Meeting*.

Matuszka, T.; Vinceller, Z.; Laki, S. (2013) On a keyword-lifecycle model for real-time event detection in social network data. In: Cognitive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on. [S.l.: s.n.] p. 453-458.

Megri, A. C.; Naqa, I.E. (2014) Prediction of the termal comfort índices using improved support machine classifiers and nonlinear kernel functions. *Indoor and Built Enviroment, 2014*. Disponível em: http://ibe.sagepub.com/content/early/2014/07/11/1420326X14539693.abstract.

Ozer, S.; Chen, C. H.; Cirpan, H.A. (2011) A set of new chebyshev kernel functions for support vector machine pattern classification. Pattern Recognition, v.44, n.7, p.1435-1447. ISSN 0031-3203.

Sakaki, T.; Okazami, M.; Matsuo, Y. (2010) Earthquake shakes twitter users: Real-time event detection by social sensors.

Salton, G.; Wong, A.; Yang, C.S. (1975) A vector space model for automatic indexing. Communications of the ACM, v. 18, p. 613-620.

Takahashi, T.; Abe, S.; Igata, N. (2011) Can twitter be an alternative of real-world sensors? *Human-Computer Interaction, Part III*, Springer-Verlag Berlin Heidelberg, p.240-249.

Turney, P.D; Pantel, P. (2010) From frequency to meaning: Vector space models of semantics. *Journal of artificial interlligence research*, AI Access Foundation and National Research Council Canada, p. 141-188.

Wang, X. et al. (2013) Real time event detection in twitter. *14th International Conference, WAIM*, Springer-Verlag Berlin Heidelberg. p. 502-513.

Weng, J. e Lee, B.S. (2011) Event detection in twitter. *Fifth International AAAI Conference on Weblogs and Social Media*, Association for the Advancement of Artificial Inteligence.