Utilizando Redes Socias para Detectar Crimes em Tempo Real

Mickael Figueredo
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte
Natal, Brasil
mickaelfigueredo@hotmail.com.br

Arthur Sousa
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
Natal, Brasil
arthurcassio@gmail.com

***Resumo*— Um dos grandes desafios do gerenciamento das cidades utilizando o conceito de Cidades Inteligentes é o alto custo para que seja possível essa abordagem. Esse artigo busca mostrar uma aplicação que utiliza uma rede social para suprir a necessidade de utilização de equipamentos de sensoriamento de elevado valor com o objetivo de detectar de crimes.**

Palavras-chave: redes sociais; processamento de linguagem natural; tempo real; twitter; machine learn

I. *INTRODUÇÃO*

O crescimento das cidades gera um aumento na complexidade de gerenciamento por parte de seus governantes. Problemas em controlar água, trânsito, segurança e saúde são muito comnuns em contextos de cidades com grande contigente populacional.

Políticos em todo mundo buscam soluções inteligentes para superar essa gama de problemas. Uma das soluções quem tem sido adotadas para superar esse desafio, é a criação de cidades inteligentes. De acordo com [1], uma cidade pode ser definida como "inteligente" quando há um investimento humano , de capital e na infra-estrutura de tecnologia da informação e comunicação(TIC).

Cidades inteligentes incorporam um grande número de sistemas, que representam a infra-estrutura mais básica para integrar o mundo real e virtual. Um dos grandes desafios para a implatação de cidades inteligentes é a extração de informações relevantes a partir da infra-estrutura TIC das cidades. Como dito em [2], tal extração, geralmente baseia-se na utilização de sensores que estão instalados para captar fluxo de veículos e pessoas, a água e o consumo de energia, necessitando assim, um alto investimento público para o desenvolvimento de cidades inteligentes.

Alguns estudos, como [3], utilizam dados de redes sociais para superar a problema de alto custo para captar os dados oriundos das cidades. A prosposta da plataforma aqui descrita, é ser capaz de detectar crimes utilizando redes socais. Para que fosse possível a criação dessa platarfoma, alguns requisitos eram necessários. No nosso caso, um sistema de aprendizagem de máquina foi criado para classificar mensagens oriundas de uma rede social, o Twitter. Além disso, toda uma estrutura de processamento em tempo real foi implementada. As seções posteriores descrevem toda essa aplicação.

II. PROCESSAMENTO DE LINGUAGEM NATURAL

De acordo com [4]. o Processamento de Linguagem Natural converte qualquer tipo de manipulação de um computador em linguagem usada para comunicação humanda. No nosso caso, o objetivo principal é analisar tweets. Para que fosse possível captar esses dados, foi utilizada uma api chamada Twitter4j[5]. Esses dados serão classificados por um classificador treinado para definir se um texto recebido no processo de captação é um provável texto falando sobre um crime.

Para que fosse possível criar esse classficador foi necessário fazer um processo de Aprendizagem de Máquina. Nesse processo, um sistema é usado para criar uma máquina capaz de aprender alguma definição a partir de um treinamento. No nosso caso, utilizamos dados de ocorrências policiais para o classificador ser capaz de definir um crime e dados oriundos do twitter para serem os dados de validação.

Uma série de sistema de aprendizagem de máquina foram testados e os mesmos possuíam diferentes resultados de acordo com o processamento de linguagem natual feito nos dados. O primeiro processamento feito é conhecido como Stemmertização. A Stemmertização[6] leva em conta o idioma de uma palavra para reduzi-la a formas não flexionadas, diretamente relacionada à uma forma de base comum, chamada de Lemma. Esse sistema ajuda a retirar ruído tanto na classificação, quanto no treinamento, assim colaborando no aumento do nível de acerto da aplicação. O Lematizador para Português[7] foi utilizado para fazer essa função na nossa aplicação.

Além disso, outro sistema para reduzir os ruídos dos nossos dados de aprendizagem e classificação é o sistema de remoção de Stop Words[4]. Esse sistema retira do texto palavras que possuem pouco valor semântico, o que reduz a quantidade de dados à serem analisados, e aumenta o peso das palavras que são mantidas nele.

Por último, cria-se o classificador da aplicação. Nessa etapa se faz uso de uma ferramenta livre chamada de Weka[8]. Com essa ferramenta, podemos criar um sistema de classificação baseado em várias métricas, que melhor se adaptam ao nosso problema. Na nossa aplicação, o treinamento e classificação são feitos utilizando um sistema de Naive Bayes, que se baseia em probabilidade de uma palavra estar mais associada à crimes ou não.

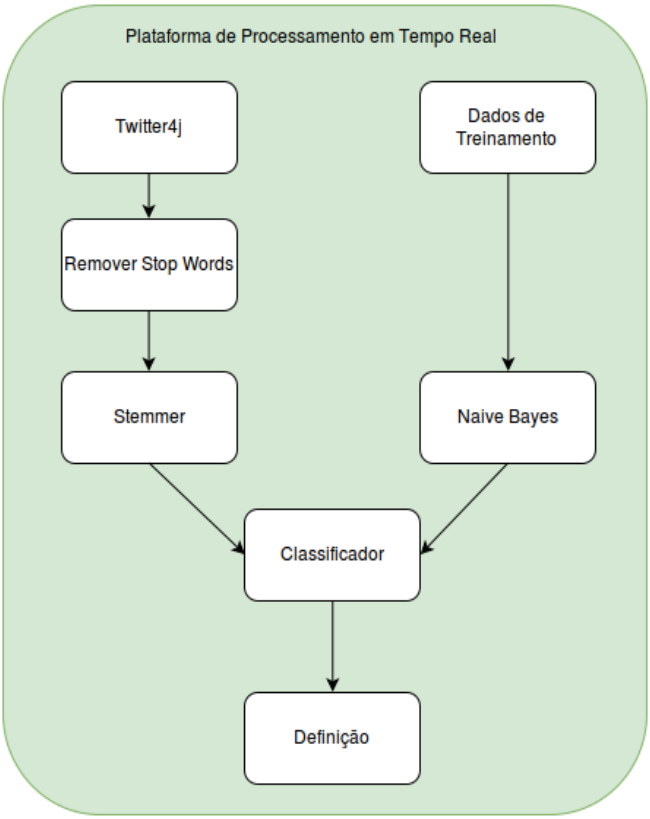
Classificador	Stemmer	Stop Words	Acerto
Naive Bayes	Sim	Não	65%
Naive Bayes Text	Não	Não	2%
Naive Bayes	Sim	Sim	98%
J48	Sim	Sim	81%

O Desempenho dos classificadores pode ser visto na Tabela 1. Podemos ver como as técnicas de processamento de linguagem natural como o Stemmer e Remoção de Stop Words colaboram para o melhor acerto do classificador. O nosso melhor desempenho utilizou todas as técnicas possíveis unidas ao classificador de Naive Bayes. Para uma classificação simples entre crime e não crime, o desempenho para classificação dos próprios dados de treinamento foi ótima, com um acerto de 98%.

III. Plataforma de Tempo Real

A plataforma utiliza infra-estrutura de processamento em tempo real como um componente central. Esta infra-estrutura é implementada através do Apache Storam[10]. O Storm é uma estrutura de processamento de fluxo livre e de código aberto capaz de processar um milhão de mensagens de 100 bytes por segundo. Um cluster do Storm é formada por uma rede distribuída de nós de processamento que processam um conjunto de dados em um compartimento de tuplas. Para isso, os três componentes definidos são Zookeper, Nimbus e o Supervisor.

O Zookeper é um serviço de alta performacer que coorderna aplicações distribuídas através da sincronização de gerenciamento de configuração, nomeação e grupo de serviços de trabalho. Na arquitetura do Storm, ele armazena a sincronização dos dados e o estado de processamento de tuplas que serão executadas nos nós do Supervisor. O Supervisor representa os nós do Storm reponsáveis por processas os dados. Finalmente o Nimbus é o nó principal do Storm, que é reposável pela distribuição de código a ser processado, atribuindo funções aos nós do Supervisor e monitorando então as falhas do processo.



A estrutura da plataforma em tempo real pode ser vista na Figura 1.

Conclusão

O artigo apresenta uma plataforma produzida para suportar uma iniciativa de cidades inteligentes. O grande objetivo dessa plataforma é captar os dados de umr rede social e retirar uma informação relevante desses dados através de processamento de linguagem natural em tempo real. Esse artigo demonstra que as redes socais podem ser utilizadas como forma de detectação de crimes. Após testes feitos, podemos dizer que os resultados foram satisafatórios. A plataforma de processamento em tempo real absorveu uma boa quantidade de dados e foi capaz de processar todos, sem presença de erros. A ferramenta foi capaz de processar certa de 2 tweets por segundo. Por outro lado, o classificador teve um bom desempenho com dados reais, porém houve um certa dificuldade quando tratávemos textos com muita ambiguidade ou teor cômico.

Referências

[1] Caragliu, A., Del Bo, C., and Nijkamp, P. (2011). Smart cities in europe. Journal of urban technology, 18(2), 65–82.

[2] Komninos, N., Pallot, M., and Schaffers, H. (2013). Special issue on smart cities and the future internet in europe. Journal of the Knowledge Economy, 4(2), 119–134.

[3] Doran, D., Gokhale, S., and Dagnino, A. (2013). Human sensing for smart cities. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 1323–1330. ACM.

[4] Bird, S., Klein, E., and Loper, E. (2009). Natural language processing with Python. ” O’Reilly Media, Inc.”.

[5] Twitter4j. A Java library for the Twitter API. Availabe at http://twitter4j.org.

[6] Classification Methods. http://www.d.umn.edu/~padhy005/Chapter5

[7] NILC, I.C.f.C.L.U. (2015). Lemmatizer for portuguese. Availabe at http://tinyurl.com/jzsxxv3.

[8] Ian H. Witten, Eiber Frank and Mark A. Hall ” Data Mining . Practical Machine Learn Tools and Techniques”.

[9] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

Article in a conference proceedings:

[10] Apache, S.F. (2015b). Apache storm. Availabe at https://storm.apache.org/.