

TrendsBot: Verificando a veracidade das mensagens do Telegram utilizando Data Stream

AUTORES

Wellison R. M. Santos¹, Marcus R. Xavier¹, David C. P. da Cunha¹, José C. F. M. Júnior¹, Daniel A. R. Aduato¹, Carlos A. G. Ferraz¹

¹Centro de Informática -- Universidade Federal de Pernambuco (UFPE)



INTRODUÇÃO

Recentemente, com o aumento de atividades em redes sociais, a facilidade da propagação de mensagens também se intensifica. Com vários canais de comunicação disponíveis, constantemente, o leitor é estimulado a propagar notícias com procedência duvidosa ou falsa. Notícias falsas ou *Fake News*, apesar de ser um assunto antigo, vêm ganhando destaque de forma recorrente, tendo seu ápice de notoriedade durante as corridas eleitorais norte-americanas de 2016 [Zhang and Ghorbani 2019].

Apesar da mobilização dos governantes e das empresas em tentar conter a propagação de notícias falsas, elas ainda se difundem nas redes sociais. Pesquisas mostram que em 2016 mais de 80% dos estudantes de nível médio tiveram problemas em distinguir notícias reais de anúncios patrocinados [Sadikoglu and Oktay 2018, Dey et al. 2018]. Em consonância com as redes sociais, os aplicativos de troca de mensagens instantâneas (IM) têm sido um dos principais disseminadores de *Fake News*.

O *Telegram* é uma aplicação de IM que possui importantes funcionalidades que foram exploradas para a elaboração deste trabalho. O *Telegram* possui a API *Bot* que possibilita que *bots* sejam implementados e/ou ligados em rede. O termo *bot* vem da palavra inglesa *robot*, que significa robô, são *scripts* que podem realizar tarefas de forma automática, como conectar com a *web*, ou até mesmo trocar informações com outros *bots* [Eslahi et al. 2013].

MATERIAIS E MÉTODOS

O *TrendsBot* (ver Figura), consiste em um *bot* que auxilia os usuários na identificação de notícias de procedência duvidosa e/ou falsa em grupos do *Telegram*. O *bot* é capaz de interagir de forma semiautomática com os membros do grupo. Uma vez mencionada a palavra-chave de acionamento "*twitter*" antecedendo a notícia de interesse (enviados através de mensagens de voz ou texto), o *bot* coleta os termos presentes na mensagem e realiza uma consulta em perfis do *Twitter* confiáveis, como o do G1 da Globo, da Corporação Britânica de Radiodifusão (BBC) e do Diário de Pernambuco. Adicionalmente, a página *web* "Fato ou Fake" é consultada para complementar as informações já coletadas nos perfis.

Para o desenvolvimento do *TrendsBot*, foi necessário um conjunto de tecnologias que seriam imprescindíveis para a integração. Sendo um *bot* para o aplicativo *Telegram*, foi necessário utilizar um cliente que pudesse implementar os comportamentos necessários quando o *bot* fosse solicitado. Logo, a biblioteca *Python Telegram Bot* foi utilizada, permitindo a adição de comandos ao *bot*. Com essa implementação, foi possível adicionar dois comportamentos fundamentais para o *TrendsBot*. O primeiro comportamento foi referente ao recebimento de um texto pelo *bot*, enquanto o segundo foi relacionado ao recebimento de áudio. Desta forma, foi programado que quando uma mensagem for enviada contendo uma palavra específica predefinida, o *bot* é engatilhado e se comunica com a aplicação servidor passando esse texto na requisição.

Após o recebimento do áudio, o fluxo de execução realiza mais uma etapa. Essa etapa é referente ao envio do áudio para a transcrição, que é realizada pela API chamada *Google Speech To Text*. Previamente, é necessário que o áudio seja enviado para o serviço de armazenamento *Google Cloud Storage*, para que de lá a API consiga transcrever o conteúdo do arquivo de áudio em questão. Se for encontrada a palavra-chave nessa transcrição, o restante do texto será enviado para ser processado pela aplicação servidor.

Após o recebimento da mensagem pelo *bot*, é executada a normalização da mensagem, onde são removidas palavras do texto que não são representativas para a compreensão geral da mensagem. Com isso, espera-se que esse processo possa gerar uma versão mais clara da

mensagem para aumentar a qualidade do resultado para outras etapas, como a de análise de sentimento. Para a normalização, foi utilizada a biblioteca NLTK (do inglês *Natural Language Toolkit*) e um repositório com palavras menos representativas da língua portuguesa.

Com o texto normalizado, a próxima etapa é realizada pelo código servidor. Para isso, o *bot* realiza uma requisição para o servidor com o conteúdo do texto normalizado. Na requisição, é feita uma consulta no *Twitter* em busca das mensagens mais relevantes sobre aquele assunto. Uma vez que o servidor coleta essas mensagens, ele inicia o processo de análise de sentimento através da API *Google Natural Language* classificando o conjunto de mensagens com os sentimentos: bom, ruim ou neutro.

O vídeo mostra uma interação sendo realizada com o *bot* através da palavra-chave `\twitter` e passando um texto para ser buscado, nesse caso a palavra "Bolsonaro". Pode-se observar no vídeo, que a palavra não está entre as "Tendências do *Twitter*" (do Inglês *Trends Topics*) e que existe uma notícia relacionada no site G1, atrelada a outra do Diário de Pernambuco. Posteriormente, é mostrado o resultado da busca por notícias que contenham o texto "Bolsonaro" no site "Fato ou Fake". Assim, os participantes do grupo podem olhar todos esses indícios, e considerá-los ao concluírem sobre a veracidade de alguma notícia, podendo ainda compartilhar esses indícios e ajudar outros contatos a refletirem se uma notícia é verdadeira ou falsa.

RESULTADOS

As notícias falsas motivam diversos pesquisadores a buscarem mecanismos e ferramentas que apontem características para auxiliarem os usuários a julgarem essas notícias como verdadeiras ou falsas. Páginas de internet como a "Fato ou Fake" são mecanismos básicos que informam quando uma notícia é verdadeira. Entretanto, é necessário que um administrador da página decida e informe ao leitor os fatos que justificam uma notícia como verdadeira ou não.

Com isso, existe a necessidade de ferramentas que automatizem a coleta de indícios para averiguar a procedência das notícias. Quando os indícios auxiliarem os usuários no julgamento da notícia, é esperado que haja uma diminuição do compartilhamento daquela notícia. Isso justifica aplicações como o *TrendsBot*, que é uma aplicação que utiliza a rede social *Twitter* e portais "Fato ou Fake" para coletar evidências que ajudem os membros de grupos de IM.

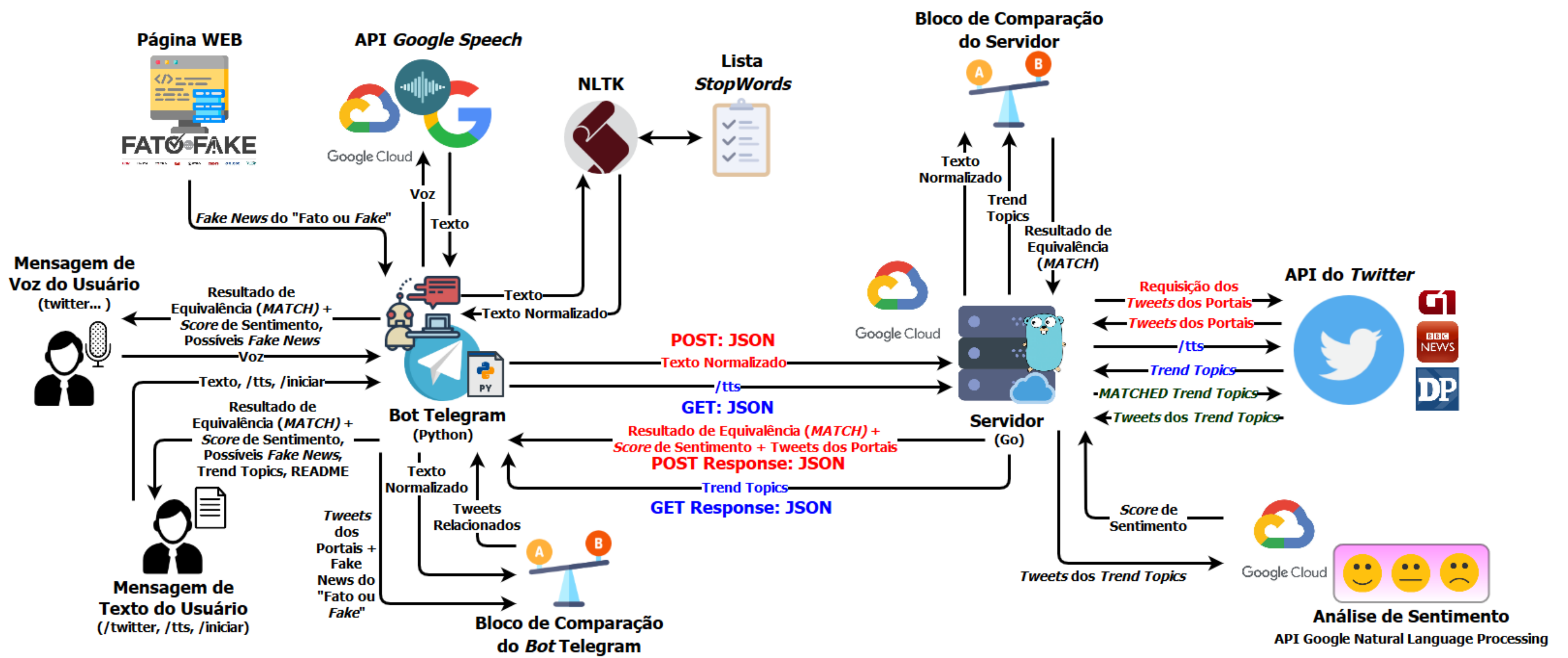
Entre os trabalhos futuros, está a busca por notícias baseadas na localização do usuário. Além disso, sempre que um link for compartilhado em um grupo, será realizada uma busca por indícios referentes a este como título e matéria. Por fim, ainda como trabalhos futuros, está a adição de mais integrações com perfis locais do *Twitter* (região, estado ou cidade) que podem trazer mais indícios no julgamento da notícia do que um perfil de abrangência nacional, e ainda a integração com mais portais do tipo "Fato ou Fake".

REFERÊNCIAS

Eslahi, M., Salleh, R., and Anuar, N. B. (2013). Bots and botnets: An overview of characteristics, detection and challenges. Proceedings - 2012 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2012, pages 349–354.

Sadikoglu, S. and Oktay, S. (2018). Identifying Fake News and Fake User s on Twitter. Procedia Computer Science, pages 204–212

Zhang, X. and Ghorbani, A. A. (2019). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, pages 1–26



Arquitetura do TrendsBot.



Vídeo de apresentação da ferramenta TrendsBot.