Analise em redes complexas da similaridade semântica dos tuítes dos deputados brasileiros

Jules Farjas

[1]Escola Politécnica – Universidade Federal do Rio de Janeiro (UFRJ)

julesfarjas@gmail.com

Abstract. *Esse artigo descreve uma exploração de um conjunto de tuítes de deputados federais brasileiros, com o ponto de vista da similaridade dos conteúdos dos tuítes. Usando técnicas de processamento de texto e elementos da ciência das redes complexas, um primeiro grafo bipartido vai ser gerado para calcular similaridade semântica, e daqui gerar um segundo grafo dos deputados com base da similaridade entre eles. Serão analisadas, por exemplo, as ligações entre conteúdo semântico e partidos políticos.*

Resumo. *This study is an exploration of a set of tweets of Brazilian deputies, from the point of view of the similarity of the contents of the tweets. Using word processing techniques and elements of complex networks science, a first bipartite graph will be generated to calculate semantic similarity, and from here generate a second graph of deputies based on the similarity between them. These results will allow us to analyze the links between semantic content and political parties, for example.*

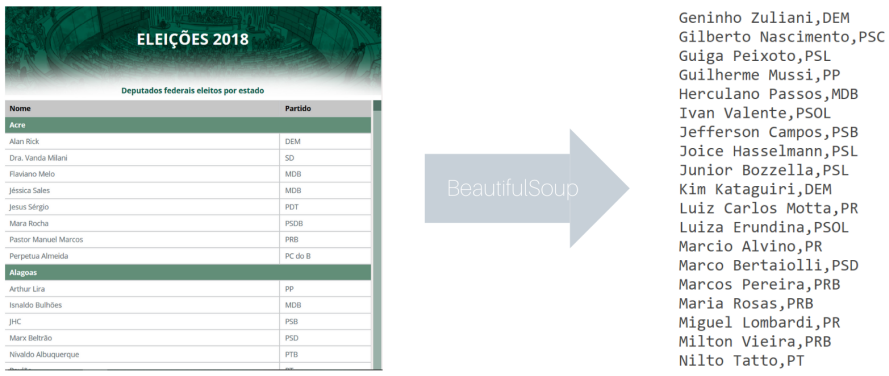
1. Introdução

Hoje em dia, os políticos usam muito as redes sociais, e particularmente o Twitter. Pensávamos que seria interessante de tentar construir uma visão geral desses tuítes, a ferramenta ideal por isso sendo uma rede. Escolhemos um ponto de vista semântico, baseado no conteúdo dos tuítes. As ferramentas usadas nesse estudo foram Python (com as bibliotecas *numpy*, *BeautifulSoup* [6], *twarc* [2], *nltk* [4] e *networkx* [3]) e o tool *Gephi* [5].

2. Recuperação dos dados

A primeira etapa é a constituição do dataset de tuítes necessário para os estudo. Essa constituição pode ser separada em 3 etapas:

2.1. Recuperar a lista dos nomes dos deputados



Primeiro, precisamos da lista dos nomes dos deputados. A lista oficial dos deputados federais por estado está disponível no site da Câmara [1]. Para a continuação do nosso estudo, queremos uma lista dos nomes (idealmente em *csv*). Por isso, escrevemos um script de scraping em Python, usando a biblioteca BeautifulSoup [6]. É so recuperar os bons marcadores HTML da pagina (tirando os nomes dos estados nesse caso). Graças a esta página, já temos os partidos de cada deputado.

2.2. Encontrar a lista das contas Twitter dos deputados

Queremos baixar os tuítes de cada deputado, então por isso temos que ter as contas Twitter deles. Para fazer isso, usáramos a API do Twitter, e mais specificamente o tool *twarc*[2], que gere automaticamente os limites de número de pedidos impostos pela API. Usando o utilitário *search*, buscamos o Twitter com cada nome recuperado acima. O jeito de escolher a pessoa certa é usando a descrição (escolher as contas verificadas pode ter sido uma boa ideia, mais todos as contas dos deputados federais não são "verificados"). Se ter a palavra "deputado"ou "deputada"na descrição, então escolhemos essa pessoa.

2.3. Baixar os tuítes dos deputados

Agora que temos a lista das contas dos deputados, temos que baixar os tuítes de cada conta. Por isso usamos também a biblioteca *twarc*, que vai baixar entre 1000 e 5000 tuítes para cada conta. Esses tuítes serão armazenados em *json*.

3. Processamento do conteúdo dos tuítes

Queremos construir um grafo bipartido bem simples: os deputados do lado esquerdo, e as palavras usadas nos tuítes no lado direito. Temos uma lista de texto simples para cada deputado, mas ainda não temos nenhuma informação sobre as características semânticas dos tuítes. Para conseguir construir esse grafo, vamos criar um dicionario das palavras usadas nesses tuítes.

3.1. Pré-processamento do texto

Primeiro, temos que extrair as palavras do texto simples. Esta é uma tarefa comum em processamento de texto, e pode ser feito em Python com a biblioteca *nltk* [4]. Temos que cortar o conteúdo em palavras, remover as palavras mais comuns que nao têm significação grande (chamadas de *stop-words*), reduzir as palavras obtidas a suas raízes (para agrupar os plurals, femininos e masculinos). Desse jeito, cada tuíte passa a ser uma sequência de palavras.

3.2. Construção dos dicionários de palavras mais usadas por deputado

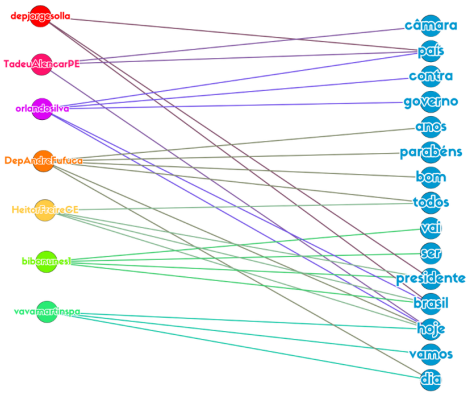
Dessas sequências de palavras, queremos extrair as palavras mais usadas por deputado (para saber a quais palavras um deputado vai ser ligado no grafo bipartido). Isso pode ser feito com um dicionário de frequência, onde cada palavra é contada e sua frequência é indicada. Desse jeito, conhecemos as palavras mas usadas de cada deputado.

3.3. Construção do dicionario geral

Para construir o dicionario geral (que na verdade sera as palavras do lado direito do grafo bipartido), é so aplicar de novo um dicionario de frequência sobre as palavras mais usadas de todas os deputados.

4. O grafo bipartido e a medida de similaridade

Cada palavra no dicionario de palavras mais usadas de um deputado é uma aresta entre esse deputado e a palavra em questão.



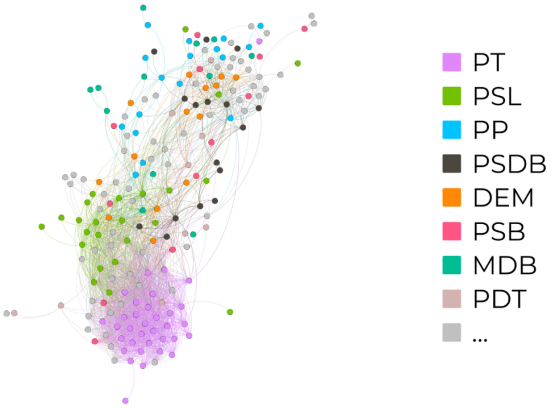
Isso permite construir uma medida de similaridade, seguindo, nosso caso, a similaridade do cosseno.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Usando essa formula e transformando cada deputado em um vetor do tamanho do dicionario geral (com um 1 se o deputado possuir essa palavra e um 0 se não), dá para obter um valor de similaridade entre cada par de deputados.

5. O grafo das similaridades

Usando esses valores de similaridade entre deputados, é possível de construir um outro grafo, essa vez, entre deputados. Uma aresta entre dois deputados sera presente se a similaridade entre esses deputados é maior do que um valor (nesse estudo, um valor de 0.5). Cada aresta tem um peso igual à similaridade, e na representação, o maior é a similaridade entre dois deputados, o mais perto eles ficam.



Para ter uma ideia mas precisa do que usando so a visualização, podemos separar o grafo em comunidades usando a medida de modularidade. Essa figura mostra uma separação dos deputados em 5 classes (com uma modularidade de 0.334), e o grafo obtido fica próximo do grafo inicial por os maiores partidos.

6. Conclusão e discussão

Parece então que, nos partidos maiores (como o PT e o PSL), os deputados têm alta semelhança semântica nos seus tuítes. Em outra nota, baseando só no conteúdo dos tuítes dos deputados, dá para reconstruir um grafo aproximado dos partidos. Esses resultados poderiam ser melhorados usando outras medidas de similaridade, e melhor processamento do texto, que ficava bem básico nesse estudo. Outros estudos semelhantes têm sido realizados, com base dos *retweets* ou *followings* por exemplo.

Referências

[1] Lista dos deputados no site da Câmara dos Deputados https://www.camara.leg.br/internet/agencia/infograficos-html5/DeputadosEleitos/index.html

[2] A biblioteca Python twarc https://github.com/DocNow/twarc

[3] A biblioteca Python networkx https://networkx.github.io/

[4] A biblioteca Python nltk https://www.nltk.org/

[5] Gephi - The Open Graph Viz Platform https://gephi.org/

[6] The scraping library BeautifulSoup https://pypi.org/project/beautifulsoup4/