IPEA - DISET

Indicador de Tecnologias Emergentes

Fred Guth*
2 de setembro de 2019

1 OBJETIVO

Antecipar tecnologias emergentes com maior potencial de impacto é essencial para um bom planejamento de políticas públicas. Usualmente, essa atividade é bastante qualitativa, se baseando no conhecimento de especialistas e na análise de patentes e artigos científicos. Entretanto, redes sociais passaram a expressar, em tempo real, a opinião de diferentes agentes econômicos, incluindo empresas, institutos de pesquisa e publicações especializadas em tecnologias emergentes; tornando possível mensurar quantitativamente, pelo número de menções, o julgamento agregado dessa multidão de agentes do potencial de tecnologias (assumindo-se que os agentes mencionam mais as tecnologias em que mais vêem potencial). É um fenômeno bem relatado (Surowiecki, 2004) que julgamentos agregados deste tipo produzem boas previsões.

Neste projeto, usamos processamento de linguagem natural para criar um indicador quantitativo para tecnologias emergentes mencionadas no Twitter por contas previamente selecionadas.

2 METODOLOGIA

O processo de construção do **Indicador de Tecnologias Emergentes (ITE)** abrange as seguintes etapas:

1. Captura dos tweets[1]
2. Análise exploratória dos dados
3. Limpeza dos dados e definição dos tópicos
4. Construção da matriz de frequências Semestre-Tópico
5. Detecção estatística de anomalias
6. Geração do Indicador

*consultor contratado.

[1] A curadoria das contas e a captura dos *tweets* foram realizadas previamente e fogem do escopo do presente projeto.

2.1 ANÁLISE EXPLORATÓRIA DOS DADOS

A análise exploratória de dados (AED) é uma abordagem de análise com objetivo de capturar um panorama geral dos dados com métodos visuais. No presente projeto, a AED não foi estensiva, até mesmo porque a maior variabilidade dos dados está no texto dos tweets o que não é capturado pela mesma. Entretanto, a AED foi importante para identificar potenciais problemas com diferentes abordagens estatísticas. Os resultados da AED serão apresentados na seção 3.

2.2 LIMPEZA DOS DADOS E DEFINIÇÃO DOS TÓPICOS

2.2.1 LIMPEZA

A limpeza dos dados tem como objetivo padronização e a remoção de termos anômalos indesejados na análise. Todas as ações de limpeza são opcionais e podem ser *desligadas* a partir de um arquivo de configuração (ver 4.3.2).

- remoção de acentos
- remoção de *hashtags* e menções a contas do twitter: apesar de hashtags serem um sinal forte de importância no Twitter, achamos importante dar a opção de fazer a análise sem hashtags.
- remoção de URLs
- remoção de números: palavras contendo apenas números, por exemplo "2017"é removido, enquanto "23andme"não é.

2.2.2 DEFINIÇÃO DOS TÓPICOS

Qualquer palavra usada em um *tweet* do corpus é, inicialmente, considerada um tópico, uma vez que, a priori, não temos como saber que uma palavra está no contexto de tecnologia. Pior do que isso, além dos termos individuais, **uni-gramas**, como analisamos também **bi-gramas**, combinações de duas palavras. No limite o número de tópicos pode ser quadrático em relação ao tamanho do vocabulário do corpus.

Tendo em vista que nosso corpus contém apenas mensagens curtas, o contexto em que termos tecnológicos aparecem não são muito identificáveis. Em outras palavras, queremos encontrar um pequeno sinal em um grande mar de ruído. Essa característica do problema nos leva a focar primeiro em identificar anomalias em geral, que podem ou não ser tecnológicas, deixando o problema da categorização dos tópicos como secundário.

Um primeiro passo é, então, reduzir o tamanho do vocabulário considerado. Parte disso é feito pela própria limpeza, mas consideramos também que palavras devem ter um número mínimo de menções-dia[2]. Além desses filtros, também estabelecemos um limite (*dict_size*) no tamanho do dicionário, pegando apenas as *dict_size* palavras com mais menções-dia no corpus.

Também filtramos *stop words*, que são palavras comuns. Todas as principais bibliotecas de NLP possuem uma lista de *stop words* do idioma inglês. Utilizamos a lista da biblioteca NLTK (Loper and Bird, 2002) e adicionamos uma lista própria de *stop words* que pode ser alterada no arquivo de configuração (§4.3.2). Tal lista foi criada especificamente para o desafio deste projeto e parte da constatação que mesmo contas especializadas engajam-se em assuntos da cultura e capturam o zeitgeist ao longo do tempo.

[2] menção-dia é frequencia de menções em diferentes dias, ou seja, várias menções no mesmo dia contam como uma.

2.3 MATRIZ DE FREQUÊNCIAS SEMESTRE-TÓPICO

Figura 2.1: Matriz de frequências Semestre-Tópico.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
antennagate	0	12	0	2	6	0	2	0	0	0	0	0	0	0	0	0	0	0	0
antennas	1	1	0	0	2	1	3	2	1	0	3	0	5	2	0	3	2	3	0
anthem	0	0	1	0	2	0	0	1	0	0	8	1	1	4	5	2	5	4	10
anthologies	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
anthology	0	0	1	1	1	0	0	0	2	0	4	7	1	12	2	8	2	1	10

A matriz semestre-tópico é uma matriz que descreve a frequencia de palavras que ocorrem semestre a semestre. É como uma matriz *document-term* onde o documento é o conjunto agregado de *tweets* de um semestre. Um dos problemas do nosso corpus que é possível verificar na Figura 3.3, é que o número de mensagens coletadas por semestre é bastante diverso e essa variação poderia afetar nosso modelo estatístico. Nesse contexto, é preciso normalizar a frequência pelo número de mensagens por semestre.

2.4 DETECÇÃO ESTATÍSTICA DE ANOMALIAS

Neste trabalho, ossa principal ferramenta para detecção de anomalias é a distribuição de Poisson, que expressa a probabilidade de uma série de eventos ocorrer em um período, dado que um evento em qualquer intervalo independe da probabilidade dele acontecer em qualquer outro intervalo.

No nosso caso, os eventos não são exatamente independentes, porque quando uma determinada conta no *Twitter* com autoridade começa a abordar um tópico, desperta o interesse de outras contas. Mas esse é justamente o caso que queremos identificar, ou seja, nossas anomalias são casos em que a distribuição de Poisson não explica a distribuição de tópicos no *Twitter* que observamos.

Seja o momento inicial das observações $t = 0$ e $N(t)$ o número de eventos que ocorrem até uma certa data t, um processo estocástico de Poisson pode ser descrito como:

$$P[N(t) = K] = \frac{e^{-\lambda t}(\lambda t)^k}{k!}, \tag{2.1}$$

onde λ representa o número de eventos por período esperado (a "velocidade"do período anterior no nosso caso). Em nossos experimentos, os primeiros 1000 uni-gramas apresentaram probabilidade de ocorrência segundo (2.1) inferior a 0.15% e no caso de bigramas, inferior a 0.03%.

2.5 GERAÇÃO DO INDICADOR

3 DADOS

Nosso corpus foi captado de uma seleção de contas no *Twitter* notórias por cobrirem assuntos de tecnologia (ver figura 3.1). Ao todo, foram coletadas 518.145 mensagens curtas entre janeiro de 2010 e 30 de junho de 2019, uma média entre 100 e 200 mensagens por dia (figura 3.2). Há uma grande variação de tweets/smestre no período, conforme é possível notar na figura 3.3.

Figura 3.1: Tweets por fonte.

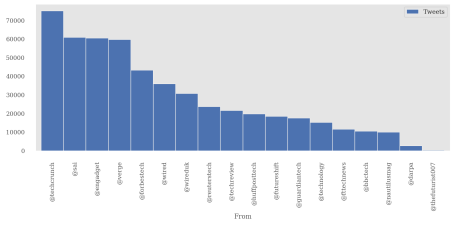


Figura 3.2: Tweets por dia.

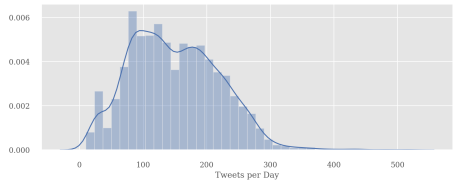
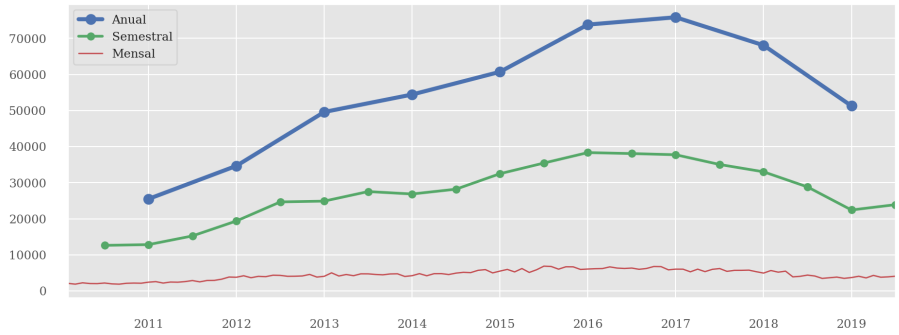


Figura 3.3: Tweets por período.



4 CÓDIGO

4.1 ACESSO

O código é aberto e seu repositório está disponível online no Github[3].

4.2 ORGANIZAÇÃO

Ao clonar o repositório, o usuário se depara com a seguinte estrutura de arquivos:

```
ipea-techmonitor
├── /data
│   └── twitter.csv
├── /docs
│   ├── data_preparation.ipynb
│   └── generate_trends.ipynb
├── config.ini
├── data_preparation.py
├── generate_trends.py
├── readme.md
├── requirements.txt
└── run.py
```

No diretório raiz encontram-se:

- **config.ini**:arquivo de configuração, onde o usuário pode alterar todos os parâmetros de execução.

[3] http://github.com/fredguth/ipea-techmonitor

- **data_preparation.py**: responsável pelo processo de limpeza dos dados, tokenização e geração de bigramas.
- **generate_trends.py**: responsável pela criação da matriz semestre-tópico, análise estatística e geração do arquivo de saída Excel.
- **readme.md**: instruções para instalação e uso do sistema.
- **requirements.txt**: lista de dependências python do projeto.
- **run.py**: pequeno programa que apenas chama **data_preparation.py** e **generate_trends.py** em ordem.

4.3 Instalação e Uso

4.3.1 Instalação

A partir do terminal de linha de comando:

```
git clone https://github.com/fredguth/ipea-techmonitor
pip install -r requirements.txt
```

4.3.2 Uso

Verifique e altere os parâmetros da execução no arquivo texto **config.ini**

Listing 1: Arquivo de configuração config.ini

```
[General]
  input_file = ./data/twitter.csv
  input_file_text_column = Tweet
  output_file = ./data/trends.xlsx
  convert_date = True
  first_date = 2010, 1, 1, 0 # datetime or nothing
  exclude_sources =
    @mashable, @techcrunch
  min_freq = 2
  max_freq = 500
  dict_size = 100000
  poisson_limit = 0.01
  trends_size = 1000
[Text Cleaning]
  actions = #use comments to prevent an action
    remove_accents,
    remove_apostrophes,
    remove_hashtags,
    remove_urls,
    remove_numberwords #words that contains only numbers
  min_word_size = 2
  max_word_size = 20
  tokenized_file = ./data/tokenized.data
```

Quando estiver satisfeito com os parâmetros, execute na linha de comando:

```
python run.py
```

REFERÊNCIAS

Loper, E. and S. Bird (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Surowiecki, J. (2004). The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business. *Economies, Societies and Nations 296*.