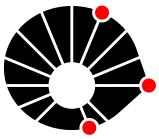
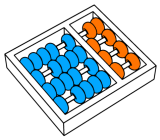




Biblioteca Aberta para Análise Temporal de Textos baseada em Modelagem de Tópicos

W. T. Ozako *A. Santanchè* *L. Gomes-Jr.*

Relatório Técnico - IC-PFG-22-42
Projeto Final de Graduação
2022 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Biblioteca Aberta para Análise Temporal de Textos baseada em Modelagem de Tópicos

Willian Takayuki Ozako * André Santanchè † Luiz Gomes Jr ‡

Resumo

Este trabalho envolveu a implementação e publicação de uma biblioteca aberta para disponibilizar técnicas de análises de discurso com modelagem de tópicos temporal. Pudemos demonstrar e validar o funcionamento da biblioteca usando casos de uso de uma pesquisa real de análise de discurso. Por fim, publicamos a biblioteca com uma licença livre e disponibilizamos-na no repositório do package manager *pip*.

1 Introdução

Este trabalho propõe a criação de uma biblioteca, aberta para a comunidade, que implemente de forma generalizável técnicas de análise temporal de tópicos. Isso permitirá a aplicação da biblioteca em cenários que necessitem extrair conhecimentos dentro de um grande volume de documentos textuais. Assim, o objetivo deste trabalho é disponibilizar um ferramental de análise de textos para futuras aplicações e pesquisas.

Para isso, desenvolvemos a biblioteca `lda-over-time`, demonstramos o seu uso e validamos a biblioteca com os valores e os resultados do artigo de Menuzzo et al. [9]. O projeto envolve um método utilizando Latent Dirichlet allocation (LDA), técnica de modelagem de tópicos, para análise de discurso. Uma de suas aplicações foi estudar como a divergência entre os discursos dos governantes, extraídos de publicações de seus perfis oficiais no Facebook, estavam correlacionados com o avanço dos casos de COVID-19 na pandemia. Ademais, publicamos a biblioteca sob licença aberta e disponibilizamos-na no repositório *PyPi* para a fácil instalação.

O restante deste trabalho está organizado da seguinte maneira: na seção 2, expomos os fundamentos utilizados neste trabalho; na seção 3, falamos sobre a organização da biblioteca e sobre os modelos implementados; na seção 4, demonstramos um caso de uso da biblioteca; na seção 5, validamos a biblioteca através da comparação do resultado encontrado na seção 4 com o resultado do artigo de Menuzzo et al.; na seção 6, levantamos os requisitos da biblioteca que implementamos neste trabalho e também levantamos os trabalhos futuros para a biblioteca.

*Instituto de Computação, Universidade Estadual de Campinas, Campinas-SP
†Instituto de Computação, Universidade Estadual de Campinas, Campinas-SP
‡Departamento de Informática, UTFPR, Curitiba-PR

2 Fundamentos

Modelagem de tópicos é um modelo estatístico que é capaz de encontrar padrões de palavras e agrupar grupos de palavras que melhor descrevem a coletânea de documentos [Pascual 2019].

A modelagem de tópicos permite então rotular documentos com suas informações temáticas em uma escala que seria inviável de ser realizada manualmente e, com isso, permite a organização e pesquisa de documentos através de seus temas [Blei 2012].

Latent Dirichlet allocation (LDA) é um modelo de modelagem de tópicos que assume que, dentro de uma coletânea, um documento é uma mistura aleatória de tópicos e cada tópico é caracterizado por uma distribuição de palavras [Blei et al. 2003]. O *LDA* assume que, para cada documento, as palavras que formam-no são escolhidas dessa maneira: 1 - escolhe-se aleatoriamente uma distribuição de tópicos; 2 - escolhe-se aleatoriamente um tópico da distribuição do passo 1; 3 - escolhe-se aleatoriamente uma palavra da distribuição de palavras do tópico escolhido em 2. Assim, para extrair a estrutura desconhecida (distribuição dos tópicos) dos documentos, realizamos o processo inverso da criação do documento. Outras suposições do *LDA* são: os documentos são representados por *bag of words*, i.e. a ordem das palavras não é importante; a ordem dos documentos não é importante (eliminando a questão temporal dos documentos); o número de tópicos é conhecida e fixa [Blei 2012].

A figura 1 ilustra um modelo de *LDA*. Os quatro tópicos da coletânea são ilustrados à esquerda, os tópicos são distribuições de palavras e são comuns a todos os documentos. Cabe ao usuário interpretar o significado de cada tópico, por exemplo o bloco amarelo aborda genética. Cada documento é considerado uma *bag of words* e encontra-se a distribuição de tópicos do documento através da atribuição de suas palavras aos tópicos existentes (histograma à direita).

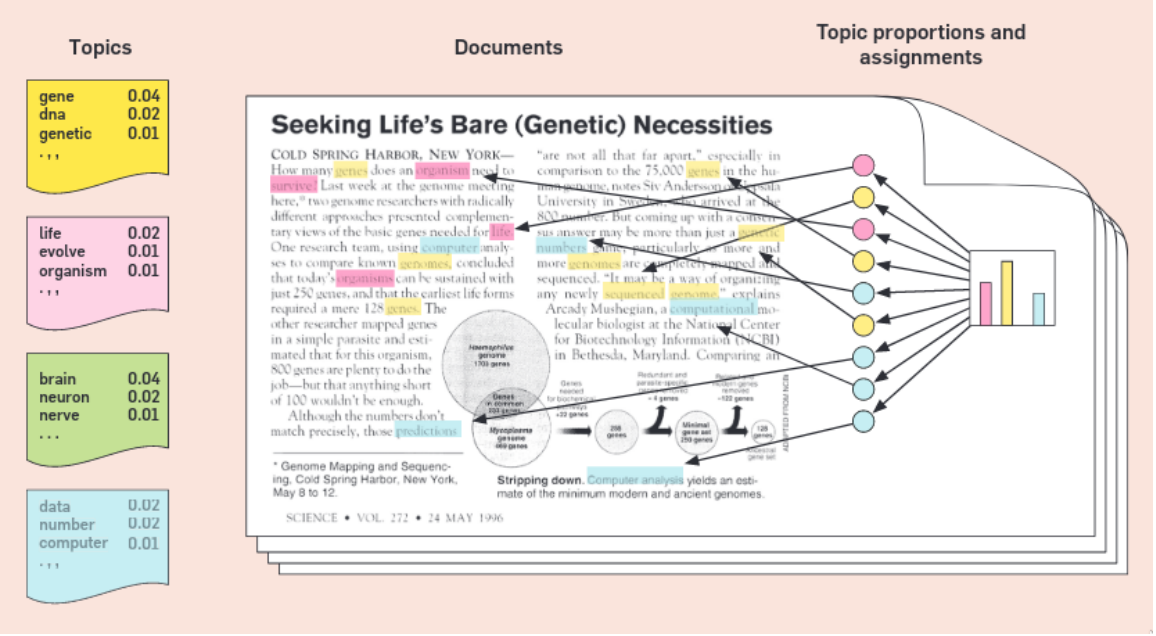
Outro uso frequente do *LDA*, nas pesquisas, envolve tópicos no tempo para estudar picos de discursos e retratar o relacionamento entre vários discursos (tópicos) no tempo. Para isso, é necessário combinar o modelo de *LDA* com outros metadados (como a marcação temporal do documento - *timestamp*) e agregar os valores. Dentre as formas de agregação há a média dos pesos de cada tópico no período e proporção de documentos que têm determinado tópico principal no período [Wieringa 2017]. A figura 3 ilustra a agregação por média e por tópicos principais de um período com dois documentos.

Dynamic Topic Models (DTM) estende o *LDA* para levar em consideração a sequência temporal dos documentos [Lee et al. 2016]. O DTM divide os documentos em fatias discretas de tempo e conecta tópicos que estão alinhados no tempo, fornecendo assim uma distribuição de termos nos tópicos que evolui no tempo [Zosa et al. 2019].

3 Sobre a biblioteca

Esta seção tem como objetivo descrever e justificar a organização dos componentes da biblioteca e também descrever o seu funcionamento.

Figura 1: Ilustrando um modelo de LDA



Fonte: Blei, 2012

O objetivo da biblioteca é disponibilizar um ferramental de análise de textos baseada em modelagem de tópicos temporal para futuras aplicações e pesquisas.

3.1 Design da biblioteca

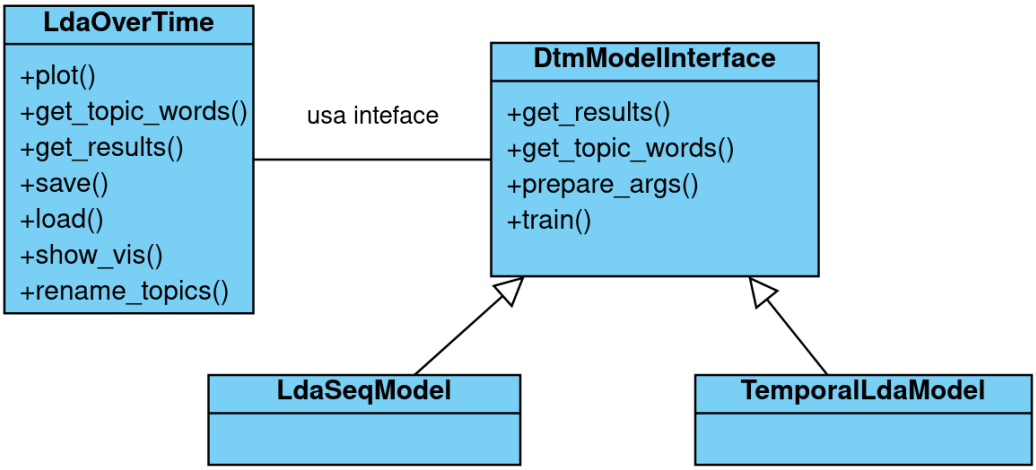
Devido ao interesse de integrar outros algoritmos que possam ser empregados na análise de discurso além do modelo descrito por Wieringa [6], decidimos pela estruturação ilustrada na figura 2. Os métodos da classe *LdaOverTime* e da interface *DtmModelInterface* estão especificados nas tabelas 1 e 2, respectivamente.

A biblioteca está dividida em duas partes principais:

- *DtmModelInterface*: é a interface requerida para que diferentes modelos implementados pela biblioteca consigam se comunicar com o componente principal da mesma, *LdaOverTime* (mesmo nome da biblioteca);
- *LdaOverTime*: implementa as funcionalidades que serão diretamente usadas pelo usuário, como ferramentais de visualização e obtenção de resultado em forma de tabela.

Para permitir maior flexibilização dos modelos que possam ser usados para estender a biblioteca, decidimos pelo fluxo onde o usuário primeiro instancia o modelo e repassa este novo objeto como argumento do componente principal, *LdaOverTime*.

Figura 2: Diagrama UML da biblioteca LdaOverTime.



Fonte: Autoria Própria

Tabela 1: Métodos de *LdaOverTime*

Método	**Descrição**
get_results	Retorna a modelagem temporal de tópicos em forma de tabela.
get_topic_words	Retorna uma lista com as n palavras mais relevantes de um tópico na fatia de tempo requisitada.
load	Restaura o estado salvo da biblioteca e retorna este estado como novo objeto da classe *LdaOverTime*.
rename_topics	Renomeia os nomes dos tópicos (o nome padrão são as 10 palavras mais importantes de cada tópico na última fatia de tempo).
save	Salva o progresso da biblioteca no arquivo especificado.
show_vis	Mostra visualmente o comportamento do modelo *LDA* treinado para a fatia de tempo especificada através da ferramenta *PyLdaVis*.

Tabela 2: Métodos de *DtmModelInterface*

Método	**Descrição**
get_results	Retorna a modelagem temporal de tópicos em forma de tabela.
get_topic_words	Retorna uma lista com as n palavras mais relevantes de um tópico na fatia de tempo requisitada.
prepare_args	Retorna os parâmetros necessários para a visualização na ferramenta *PyLdaVis* na fatia de tempo requisitada.
train	Método chamado para acionar o treinamento do modelo temporal de tópicos. Não retorna valores.

3.2 Modelos implementados

Nesta subseção, apresentamos os modelos implementados para a biblioteca.

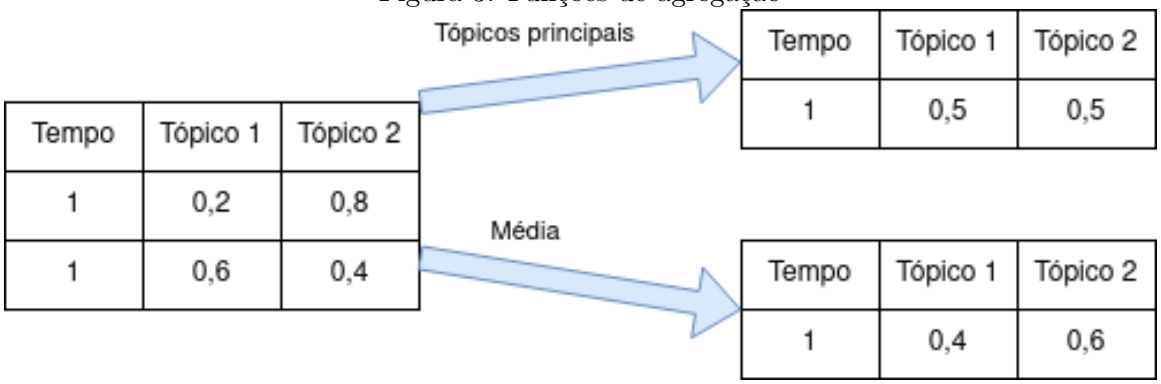
Relembrando que todos os modelos devem implementar a interface *DtmModelInterface*, ilustrada pela figura 2 junto com a tabela 2.

3.2.1 TemporalLdaModel

O modelo *TemporalLdaModel* implementa a técnica descrita por Wieringa [6], isto é, ele treina um modelo de LDA e agrupa documentos que pertencem à mesma fatia de tempo através de uma função agregadora. Implementamos a média e a proporção de tópicos principais. A biblioteca usada para o LDA é *Gensim*[1] com o modelo `gensim.models.ldamulticore.LdaMulticore` e a redução foi feita com *Pandas*[2].

Na figura 3, ilustramos a redução de um modelo de LDA (à esquerda) para o modelo temporal de tópicos (à direita) através de duas funções distintas: na primeira redução, vemos que um documento tem como tópico principal o tópico 2 e o segundo tem o 1 como principal, por isso o resultado é 50% para ambos os tópicos; na segunda redução, aplicamos a média nos pesos de cada tópico para os documentos da mesma fatia de tempo, resultando em 40% e 60%.

Figura 3: Funções de agregação



Fonte: Autoria Própria

3.2.2 LdaSeqModel

Devido a característica do *DTM* em detectar mudanças temporais na forma em que os tópicos são descritos, decidimos adicionar o suporte para o modelo `gensim.models.ldaseqmodel.LdaSeqModel` do *Gensim*[3], que implementa o modelo de *DTM*.

[1] https://radimrehurek.com/gensim/index.html
[2] https://pandas.pydata.org/
[3] https://radimrehurek.com/gensim/index.html

Blei [3] exemplifica um caso de dois artigos, um datado em 1903 e outro em 1991, que fazem parte da mesma trajetória científica, mas devido à mudanças na maneira que o campo da neurociência aborda o assunto, estes artigos parecem pertencer a campos distintos. Neste exemplo, o *LDA* provavelmente iria atribuir tópicos distintos para os dois artigos, enquanto o *DTM* iria encontrar um caminho que conectasse os dois artigos temporalmente e atribuir tópicos comuns para os dois artigos.

4 Demonstração de um caso de uso da biblioteca

Nesta seção, demonstramos um caso de uso da biblioteca. Utilizamos os mesmos dados e procedimentos descritos no artigo de Menuzzo et al [9]. para posteriormente validarmos a biblioteca.

O artigo de Menuzzo et al. realizou análises de discurso nas postagens pertencentes às prefeituras e aos prefeitos das capitais brasileiras. Os autores propuseram formas de análises de discurso utilizando modelagem de tópicos e métricas de análise como a diversidade e coesão. Com estes métodos, eles encontraram uma correlação entre a distância do discurso central e a mortalidade por Covid-19.

4.1 Origem e preprocessamento dos dados

Os dados utilizados são os mesmos do artigo de Menuzzo et al. Estes dados foram extraídos no período de 2020 até 2021 e são postagens pertencentes às prefeituras e aos prefeitos das 26 capitais brasileiras na rede social Facebook. Somente contas oficiais foram utilizadas.

Para garantir melhores resultados o usuário deve realizar o preprocessamento dos dados com bibliotecas externas, visto que a nossa biblioteca não tem suporte para o processamento.

Assim como no artigo mencionado acima, para selecionarmos as postagens que estão relacionadas a COVID, selecionamos os textos que usam pelo menos um dos vocábulos destacados pelo de Melo [8]. E, por fim, removemos *stop words*, *links* e aplicamos a lematização das palavras com o uso da biblioteca *spaCy*[4].

4.2 Escolha do modelo

Como o artigo [9] utilizou uma modelagem de tópicos baseada em *LDA* e utilizou a média como função agregadora, escolhemos o modelo que implementa o mesmo algoritmo do artigo: *TemporalLdaModel* configurado para usar a média como função agregadora.

Abaixo, instanciamos o modelo escolhido com os valores necessários: lista com os textos; lista das datas de postagem; formato da data fornecida, AAAA-MM-DD; agrupar em períodos de um mês; extrair cinco tópicos; usar a média como função redutora. A sequência de textos na lista `corpus` deve corresponder a sequência de datas na lista `dates`.

[4] https://spacy.io/

```
from lda_over_time.models.temporal_lda_model import TemporalLdaModel

model = TemporalLdaModel(
  corpus=corpus,              # list of texts
  dates=dates,                # list of dates
  date_format="%Y-%m-%d",     # date format used in `dates`: YYYY-MM-DD
  freq="1M",                  # split in groups of one month
  n_topics=5,                 # get 5 topics
  aggregator="average"        # use average
)
```

4.3 Treinando o modelo

Para treinar o modelo acima, repassamos o objeto do modelo para o componente principal da biblioteca. A partir deste ponto, o usuário precisa interagir somente com o novo objeto *main*.

```
from lda_over_time.lda_over_time import LdaOverTime

main = LdaOverTime(model)
```

4.4 Atribuição do significado de cada tópico

O método `get_topic_words` é usado para obter as n palavras mais importantes do tópico escolhido na fatia de tempo escolhida. No código abaixo, pegamos as primeiras 15 palavras, da primeira fatia de tempo de cada um dos tópicos, que são numerados de *1* até *5*, e obtemos o resultado na tabela 3.

```
for topic_id in range(1, 6):
  top15 = ', '.join(main.get_topic_words(
    topic_id=topic_id, # select topic by id
    timeslice=1,       # get key words from first time slice
    n=15               # get top 15 words
  ))
  print(top15)
```

A partir da tabela 3 aferimos que os tópicos são, respectivamente: alertas gerais; vacinação; novos casos; prevenção; atendimentos (hospitais e leitos).

4.5 Plotando o gráfico

Antes de plotar, renomeamos os tópicos através do método `rename_topics`, passando como argumento uma lista com os novos nomes. Caso esse método não seja chamado antes de gerar o gráfico, os tópicos serão nomeados pelas suas respectivas 10 palavras-chaves mais significativas.

Tabela 3: Top 15 palavras de cada tópico

Id	**15 palavras mais relevantes**
1	caso, coronavirus, obito, confirmar, covid-19, boletim, saude, suspeito, novo, paciente, registrar, logo, acessar, atualizar, epidemiologico
2	ano, vacinacao, dia, idoso, ser, dose, contra, saude, covid-19, vacina, pessoa, vacinar, obito, estao, primeiro
3	caso, novo, covid-19, capital, leito, saude, belem, coronavirus, boletim, acre, municipal, numero, ser, registrar, confirmar
4	coronavirus, ser, medida, voce, ter, casa, pandemia, pessoa, covid-19, novo, poder, evitar, como, social, mascara
5	saude, ser, municipal, atendimento, covid-19, unidade, dia, novo, coronavirus, pandemia, realizar, secretaria, rede, hospital, servico

```
main.rename_topics(topics)
```

Para plotar o gráfico, temos o método `plot` e passamos os seguintes parâmetros: título do gráfico; nome da legenda; local para salvar o gráfico (se não especificado o gráfico não é salvo); rotação dos valores das datas em 75°; plotar um gráfico de áreas empilhadas (o padrão é plotar um gráfico de linhas); formato de data para aparecer no gráfico. O gráfico resultante está na figura 4.

```
main.plot(
  title = "Analise de posts sobre Covid-19", # graph's title
  legend_title="Topicos",                    # legend's title
  path_to_save="./covid-stack.png",          # save at
  rotation=75,                               # rotate dates in 75o
  mode="stack",                              # use stack plot
  date_format="%m/%Y",                       # display dates in MM/YYYY
)
```

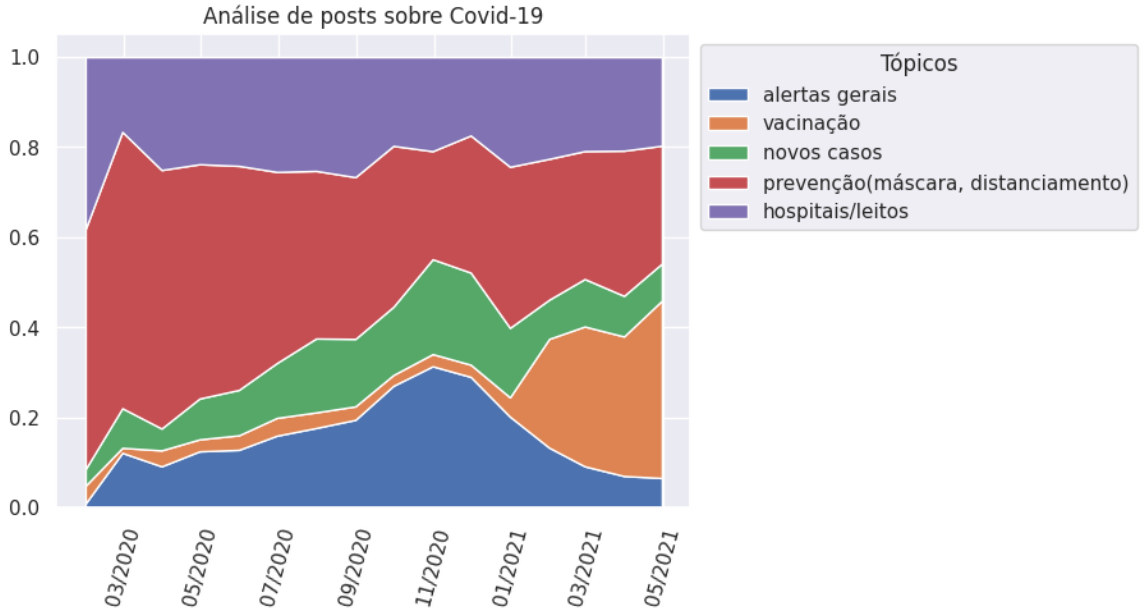
4.6 Analisando o modelo treinado

O último método a ser explorado aqui é o `showvis`, que usa a biblioteca *PyLdaVis* para gerar a visualização do modelo de LDA e averiguar a qualidade do modelo treinado. O nosso método recebe apenas o valor da posição da fatia de tempo desejada (numerada de *1* a *20*), no exemplo temos a análise de março de 2020. O resultado da análise é apresentado na figura 5.

```
main.showvis(1)
```

Detalhamos mais sobre os modelos e métodos da biblioteca na documentação https://lda-over-time.github.io.

Figura 4: Proporção dos tópicos



Fonte: Autoria Própria

5 Discussão

Comparando os resultados obtidos na figura 4 com o resultado apresentado por Menuzzo et al. [9], podemos ver que os resultados são parecidos. E, com isso, validamos que a biblioteca funciona corretamente.

Como exemplo, segundo Menuzzo et al., houve um primeiro momento em que o foco das postagens era alertar e dar informações à população e, em outro período, houve uma grande ênfase sobre a vacinação. Este comportamento relatado pode ser observado na figura 4.

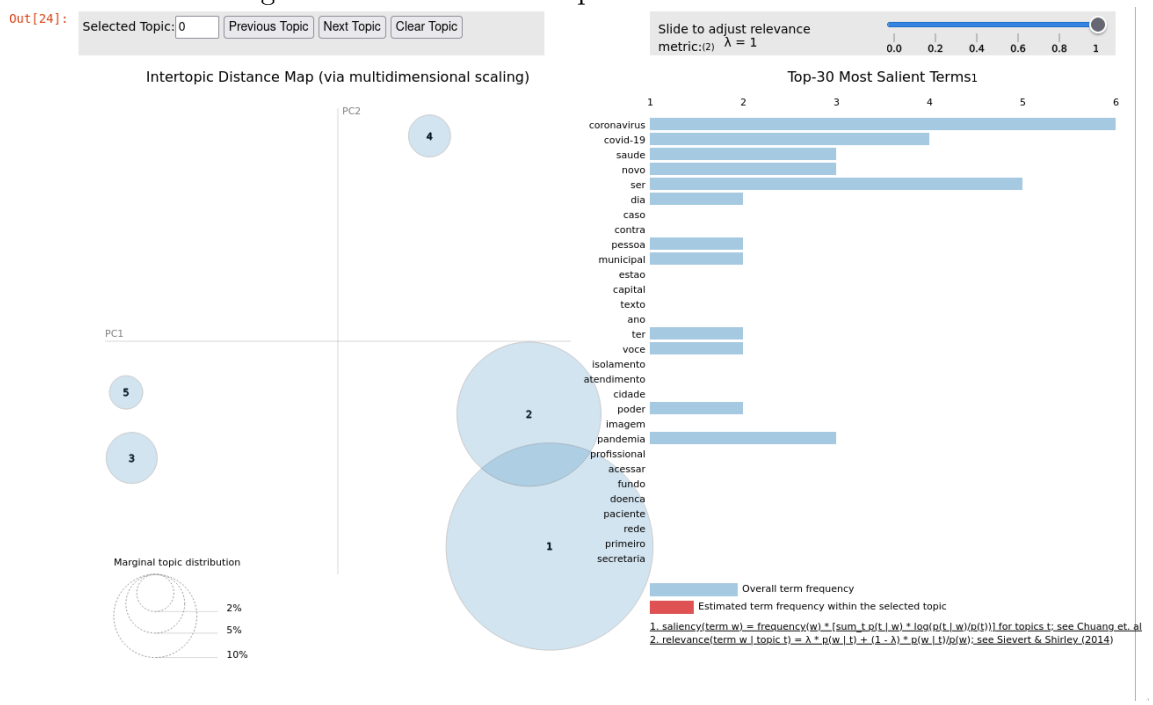
Houve porém algumas diferenças nos valores encontrados de cada mês, como em maio de 2020. Enquanto o nosso modelo apontou que 50% das postagens foram sobre prevenção, Menuzzo et al. encontraram um valor de aproximadamente 40%. Contudo, essa divergência deve-se a algumas diferenças no preprocessamento, como a definição das *stop words*. Por exemplo mantivemos os nomes das cidades e é possível que Menuzzo et al. tenha eliminado.

6 Conclusão

O objetivo da biblioteca é disponibilizar um ferramental de análise de textos baseada em modelagem de tópicos temporal para futuras aplicações e pesquisas.

Desenvolvemos a biblioteca de maneira que esta fosse facilmente expansível com a inclusão de novos modelos de modelagem temporal e, atualmente, temos dois modelos dis-

Figura 5: Avaliando desempenho do modelo treinado



Fonte: Autoria Própria

poníveis para o uso. Demonstramos na seção 4 que as funcionalidades da biblioteca são fáceis de serem usadas e conseguimos plotar os gráficos de evolução dos tópicos no tempo, com poucas linhas de código, e também averiguar a qualidade do modelo treinado com a integração da biblioteca *PyLdaVis*. Como trabalho futuro, planeja-se a implementação das métricas defendidas por Menuzzo et al., como a coesão e a diversidade de discursos.

Também criamos um formulário para averiguar a facilidade de uso da nossa biblioteca. Questionamos sobre: facilidade de instalação; facilidade de aprendizado através dos materiais fornecidos (tutorial e documentação); se encontraram algumas limitações no uso da biblioteca (a biblioteca não é genérica o suficiente); se encontraram algum *bug*. Porém não houve tempo o suficiente para realizar esta validação, mas pretendemos realizá-lo para levantar melhorias na biblioteca.

Por fim, a biblioteca está disponível com sob a licença *GNU Lesser General Public License v3.0* (*LGPL-3.0*) no repositório
https://github.com/lda-over-time/lda-over-time. Além disso, para facilitar a instalação e uso da ferramenta, publicamos a framework em *Pypi* e disponibilizamos a documentação em https://lda-over-time.github.io.

Referências

[1] D. M. Blei. *Probabilistic topic models. Commun.* ACM 55, 04 abr. 2012, 77–84. https://doi.org/10.1145/2133806.2133826.

[2] D. M. Blei, A. Y. Ng, e M. I. Jordan. *Latent dirichlet allocation. Journal of machine Learning research 3*, Janeiro 2003: 993-1022.

[3] D. M. Blei e J. D. Lafferty. *Dynamic topic models. In Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery*, New York, NY, USA, 113–120, 26 jun. 2006. https://doi.org/10.1145/1143844.1143859

[4] E. Zosa e M. Granroth-Wilding, *Multilingual Dynamic Topic Model . in G Angelova , R Mitkov , I Nikolova e I Temnikova (eds) , RANLP 2019 - Natural Language Processing a Deep Learning World : Proceedings . International conference Recent advances in natural language processing , INCOMA , Shoumen , Recent Advances in Natural Language Processing* , Varna , Bulgaria, 1388-1396, 02 set. 2019. https://doi.org/10.26615/978-954-452-056-4_159

[5] F. Pascual. *Topic Modeling: An Introduction. MonkeyLearn*, 26 set. 2019. Disponível em: https://monkeylearn.com/blog/introduction-to-topic-modeling. Acesso em: 11 dez. 2022.

[6] J. E. Wieringa. *Ways to Compute Topics over Time, Part 1. Jeri E. Wieringa*, 21 jun. 2017. Disponível em: https://jeriwieringa.com/2017/06/21/Calculating-and-Visualizing-Topic-Significance-over-Time-Part-1/. Acesso em: 04 dez. 2022.

[7] M. Lee, Z. Liu, R. Huang e W. Tong. *Application of dynamic topic models to toxicogenomics data. BMC Bioinformatics*. 06 out. 2016;17(Suppl 13):368. doi: 10.1186/s12859-016-1225-0. PMID: 27766956; PMCID: PMC5073961.

[8] T. de Melo e C. M. S. Figueiredo. *A first public dataset from Brazilian twitter and news on COVID-19 in Portuguese. Data Brief.* 2020;32:106179. doi:10.1016/j.dib.2020.106179

[9] V. Menuzzo, A. Santanchè e L. Gomes-Jr. *Evaluating the cohesion of municipalities' discourse during the COVID-19 pandemic. In Anais do XXXVI Simpósio Brasileiro de Bancos de Dados*, 04 out. 2021, Rio de Janeiro, Brasil. SBC, Porto Alegre, Brasil, 295-300. DOI: https://doi.org/10.5753/sbbd.2021.17888.