# From audio to information: Learning topics from audio transcripts

João Pedro Santos Rodrigues, Emerson Cabrera Paraiso

Pontifícia Universidade Católica do Paraná, Brazil
pedro.santos@ppgia.com.br, paraiso@ppgia.pucpr.br

**Abstract.** In this work, the technical feasibility of working with audio transcriptions from Youtube is analyzed, as well as presenting a method that allows data acquisition, pre-processing, and post-processing to work with this type of data. A topic modeling approach with the latent dirichlet allocation algorithm is used. An approach is also presented to dynamically determine the ideal number of topics that make up a given corpus. In the experiments, a database of 250 audio transcriptions was used, obtaining a model with coherence in the range of 40%.

CCS Concepts: • **Applied computing**;

Keywords: audio transcription, data mining, machine learning, topic modeling

## 1. INTRODUÇÃO

Uma das redes mais sociais mais utilizadas no mundo é o Youtube. Basicamente esta rede é uma plataforma de criação e compartilhamento de vídeos gratuita onde são assistidas mais de 2 bilhões de horas de vídeos diariamente [Youtube 2020]. Consequentemente, um grande volume de dados é produzido diariamente, sendo rapidamente disponibilizado e compartilhado entre os usuários. O grande volume de dados produzido, bem como sua rápida disseminação, resultam na impossibilidade de realizar a curadoria dos dados através de processos manuais [Patel et al. 2012]. Este fato aumenta o risco da publicação conter informações inverídicas e errôneas; além disso, do ponto de vista do usuário, o processo de encontrar a informação que ele deseja também é custoso [Gausby 2015]. Desta forma, abre-se uma oportunidade para o emprego de técnicas ou ferramentas que permitam a análise e a interpretação dos conteúdos publicados nesta rede social.

Diversas pesquisas foram propostas para a aquisição de conhecimentos a partir de redes sociais baseadas em vídeos. Nos trabalhos de [Kaushik et al. 2013], [Rangaswamy et al. 2016] e [Wöllmer et al. 2013], por exemplo, são apresentadas diferentes formas para análise de sentimentos a partir do Youtube; porém, apesar dos autores utilizarem uma rede social baseada em vídeo como fonte de dados em seus processos, tais pesquisas objetivaram o desenvolvimento ou de ferramentas de análise de sentimento ou de sistemas de recomendação.

Os autores em [Munaro et al. 2020] apresentaram um modelo de obtenção de conhecimento em redes sociais baseada em vídeo utilizando transcrições de áudio e metadados dos vídeos, mas apenas focada na descrição da popularidade dos vídeos. Já os autores em [de Souza and Souza 2019], propuseram um modelo de obtenção de conhecimento através da análise e interpretação de conteúdos objetivando a representação deste conteúdo; porém, o método proposto é dependente da análise de um especialista humano, o que torna o processo inviável (ou extremamente custoso), se aplicado ao grande conjunto de vídeos produzidos diariamente pelas plataformas de vídeo. Em outras palavras,

há poucos trabalhos na literatura que explorem a aquisição de conhecimento em redes sociais baseadas em vídeo de maneira automatizada, ou que realizem análises sintática, morfológica e/ou pragmática dos conteúdos componentes de um vídeo. Neste trabalho, apresentamos um método para a extração de conhecimento de vídeos de redes sociais por meio de transcrições de áudio destes vídeos. Este método inclui as etapas de aquisição, pré-processamento, e pós-processamento dos dados, com o objetivo de sintetizar e agrupar diferentes documentos (transcrições) a partir de seus assuntos componentes. Para isto, é utilizado uma abordagem de modelagem de tópicos com o algoritmo *latent dirichlet allocation* (LDA) [Blei et al. 2003].

As transcrições de áudio (no caso do Youtube) podem ser geradas de duas formas diferentes: (i) a partir da anotação manual áudio[1]; e (ii) a partir de algoritmos que façam o reconhecimento automático da fala (*automatic speech recognition* (ASR))[2]. O texto gerado a partir do ASR pode possuir ruídos, como erros sintáticos e semânticos; com isso, o uso deste dado bruto em um sistema de aquisição de conhecimento pode ser prejudicado. Além desta dificuldade, um dos grandes desafios ao se trabalhar com modelagem de tópicos é encontrar o número "ideal" de tópicos em um dado corpus (conjunto de documentos de entrada do processo de modelagem de tópicos), já que diferentes tópicos representam o conhecimento de maneiras diferentes. Neste sentido, também propomos uma abordagem para a inferência do número de tópicos a partir de uma análise de coerência do modelo. Por fim, o método também tem como objetivo encontrar os tópicos que compõem o corpus utilizado, relacionando estes documentos entre os domínios (tópicos) e entre os canais.

O restante deste artigo está organizado da seguinte forma: na seção 2, são apresentados os trabalhos relacionados e uma breve fundamentação teórica do tema. A seção 3 apresenta o método proposto neste trabalho. Na seção 4 são apresentados os experimentos realizados e os seus resultados. Por fim, a última seção apresenta a conclusão e possíveis trabalhos futuros.

## 2. TRABALHOS RELACIONADOS E FUNDAMENTAÇÃO TEÓRICA

Diversos trabalhos na literatura tratam a aplicação de técnicas de processamento de linguagem natural e extração da informação em dados obtidos em redes sociais, como por exemplo utilizando o LinkedIn [He et al. 2016], Facebook [Noy et al. 2019], Twitter e outras. Porém, conforme apresenta Alexe e colegas [Alexe et al. 2012], diferentes mídias sociais, possuem diferentes características. Por exemplo: postagens do Twitter (tweets) apresentam um estilo mais direto e focado. Em contra partida, estes textos tendem a ser mais ruidosos (como uso massivo de abreviações, emojis e hashtags). Já textos oriundos de blogs, possuem menos ruídos, entretanto, as informações relevantes estão dispersas em uma maior quantidade de palavras. Já Wollmer e colegas, demostram que transcrições de áudio possuem um caráter coloquial com uma grande quantidade de gírias, neologismos e abreviações [Wöllmer et al. 2013]. Adicionalmente a isto, as transcrições obtidas a partir de algoritmos de ASR, possuem uma completa falta de pontuação e erros semânticos.

No trabalho de [de Souza and Souza 2019], os autores apresentam uma análise entre os modelos de tópicos a partir do algoritmo LSI (latent semantic indexing) e LDA (latent dirichlet allocation). Para a geração dos modelos os autores utilizaram uma abordagem incremental no número de tópicos em cada modelo. Ao fim deste processo cada modelo é avaliado por um especialista a fim de determinar qual modelo melhor descreve os tópicos a partir de suas palavras chave. Os autores concluem que, modelos gerados pelo algoritmo LDA, tendem a gerar resultados mais interpretáveis aos humanos, ao passo que o LSI gera tópicos com palavras mais diversas.

Tanto o LSI como o LDA, constituem uma família de algoritmos com o objetivo de realizar a modelagem de tópicos. A modelagem de tópicos é o campo de mineração de texto e recuperação da informação com o objetivo de extrair tópicos latentes de um corpus. Entende-se como um tópico

[1]Youtube help. Disponível em <https://support.google.com/youtube/answer/2734796?hl=en>
[2]Youtube help. Disponível em <https://support.google.com/youtube/answer/6373554?hl=en>

latente (ou escondido), um conjunto de termos ou palavras que permita caracterizar um conjunto de documentos como semelhantes ao mesmo passo que possibilite distingui-los dos demais documentos de um corpus. Desta forma, é possível classificar textos e documentos de maneira não supervisionada, para encontrar assuntos ou domínios semelhantes entre documentos diferentes [Liu et al. 2016]. Apesar de parecer uma tarefa "fácil" de ser realizada por humanos, esta tarefa é complexa em termos computacionais, uma vez que o computador desconhece os significados dos termos ou palavras em tempo de execução. Uma analogia para compreendermos o tamanho deste desafio, seria pedir para uma pessoa determinar os assuntos tratados em um conjunto de documentos em um idioma desconhecido.

O LDA é um dos algoritmos que possuem os melhores resultados no estado da arte, gerando modelos com uma boa generalização dos dados, principalmente quando se trabalha com um grande volume de dados [Blei et al. 2003] [Misra et al. 2009]. Este algoritmo foi introduzido por [Blei et al. 2003] e seu funcionamento é baseado na geração de um modelo generativo probabilístico bayesiano, onde cada documento é tratado como uma mistura aleatória de tópicos latentes. Já os tópicos são tratados como uma distribuição de palavras ponderadas pelo seu peso. Desta forma, primeiramente o modelo define K tópicos (passado como parâmetro), onde cada k tópico é associado a uma distribuição $\psi_k$ sobre as palavras presentes no vocabulário do corpus. Esta distribuição $\psi_k$, é obtida através da distribuição de Dirichlet $\beta$. De posse dos tópicos criados, cada documento d (da coleção de palavras $w_d$) é gerado a partir distribuição $\theta_d$ nos K tópicos através da distribuição ${}_k(\alpha)$. Com isso será possível determinar o peso de cada palavra $w_{di}$ em cada tópico presente no conjunto $w_d$ através de uma distribuição $\theta_d$. Por fim o LDA seleciona os tópicos $z_{di} \in [1, K]$ através de uma distribuição multinomial ${}_k(\theta_d)$. Finalmente cada palavra $w_{di}$ é selecionada através da distribuição ${}_v(\psi_{z_{di}})$. Como limitação, O LDA possui a exigência de que o número de tópicos seja previamente definido como parâmetro na geração do modelo, o que limita a aplicabilidade do LDA em diversos cenários, principalmente nos casos de corpus de domínio aberto.

Do ponto de vista da avaliação dos modelos de tópicos latentes, podemos avaliá-los sob três óticas diferentes: (i) especialista humano, (ii) perplexidade e (iii) coerência. Segundo [Hagen 2018], modelos gerados com uma alta coerência tendem a gerar tópicos com uma maior interpretabilidade pelos leitores humanos. Assim, é recomendada a utilização de modelos com uma alta coerência. O cálculo da coerência pode ser realizado por meio de uma função de pontuação (ou *score*) de similaridade semântica entre termos no formato $\sum_{i<j} score(w_i, w_j)$ [O'callaghan et al. 2015]. Esta função de *score* pode ser calculada de diversas formas dependendo do domínio do problema ao qual o modelo é aplicado. Dentre as medidas existentes na literatura, destaca-se a medida de associação NPMI (*Normalized Pointwise Mutual Information*) [Syed and Spruit 2017]. A medida de *score* NPMI é descrita por $1/{}^{N}C_2 \sum_{j=2}^{N} \sum_{i=1}^{j-1} ( log\ (P(w_j, w_i) + \epsilon / P(w_j)\ P(w_i)) / - log(P(w_j,\ w_i)) + \epsilon)$.

## 3. MÉTODO PROPOSTO

O método apresentado nesta seção possui as seguintes etapas, descritas a seguir: (i) aquisição dos dados, (ii) pré-processamento e (iii) obtenção de tópicos latentes e seleção do melhor modelo.

Devido à ausência de um corpus disponível na literatura com transcrições de áudio a partir de vídeos do Youtube, foi desenvolvida uma ferramenta para este fim. Para a obtenção das transcrições de áudio foi utilizado um *web crawler* com o objetivo de coletar os arquivos de texto das transcrições. Os demais dados (título, descrição, url, ...), foram coletados utilizando a API oficial disponível (chamada de Youtube Data V3).

Para treinar o modelo para a obtenção dos tópicos latentes é importante realizar uma série de pré-processamentos no corpus. O primeiro passo consiste em realizar operações básicas como a remoção das *stop words* e transformar todos os caracteres maiúsculos em minúsculos. Com o objetivo de diminuir a dimensionalidade dos dados, aplica-se também um lematizador. Adicionalmente, são identificados todos os bi-gramas e tri-gramas presentes no corpus. Esta tarefa é importante pois a obtenção dos

n-gramas também permite a diminuição da dimensionalidade dos termos, ao passo que aumenta a sua representatividade. Consequentemente, isto tende a melhorar o resultado dos modelos gerados [Blei et al. 2003]. Dois parâmetros são importantes para o cálculo do n-gramas: a frequência mínima de ocorrência das palavras (*min_count*) e o limite de corte (*threshold*). O *min_count* é responsável por determinar quais termos são elegíveis para formar os gramas. Só são aceitos termos que possuem uma frequência maior que este parâmetro. Após a geração dos bi e tri-gramas o *threshold* determina quais os gramas serão mantidos na base processada. Todo grama gerado que tenha uma frequência menor que o *threshold* é descartado. Após testes empíricos, determinou-se um *threshold* de 50 e um *min_count* de 5. Por fim, para se obter uma representação vetorial dos termos e n-gramas da base, foi gerado o *bag-of-words* da base a partir da contagem dos termos presentes.

Neste trabalho foi utilizado o algoritmo de extração de tópicos latentes LDA. Este algoritmo possui como limitação a impossibilidade de inferir o número ideal de tópicos de um corpus (chamado de $k$). Desta forma, este número deve ser definido antes da sua execução. Uma abordagem comum para inferir o número ideal de tópicos, é gerar diversos modelos com diferentes valores de $k$ e com o auxílio de um especialista analisar os resultados obtidos, até encontrar um resultado satisfatório [de Souza and Souza 2019]. Desta forma, foi necessário desenvolver uma abordagem que permitisse encontrar o melhor número de $k$ em um corpus de forma automatizada, conforme demostrado na figura 1.

O algoritmo 1 apresenta o processo que está sendo proposto para a obtenção automática do número "ideal" de tópicos. O método gera um conjunto de $n$ modelos de forma a ir incrementando o número de $k$ tópicos. Após a criação de cada modelo, é também calculada coerência deste modelo a partir da média da coerência dos tópicos. Este cálculo será importante por duas razões: (i) determinar o critério de parada na geração de novos modelos e (ii) auxiliar na seleção do melhor modelo. Para determinar o critério de parada das geração dos modelos incrementais, é calculado o coeficiente de correlação de Pearson entre a coerência dos modelos e o seu número de tópicos. Este valor será responsável por determinar se as coerências dos modelos seguem uma tendência alta ou queda em relação ao incremento dos k tópicos. Desta forma o cálculo da correlação será feito a partir de janelas deslizantes (*jd*) pré-definidas. A partir do momento que for detectada uma correlação negativa (tendência de queda) nos modelos que compõe a jd, o treinamento se encerra.

A próxima etapa a ser executada, é selecionar o candidato a melhor modelo da lista dos modelos obtidos e que consequentemente possua o número mais adequado de tópicos que represente o corpus utilizado. Isso é feito selecionando o modelo com a maior coerência disponível. Por fim a ultima etapa a ser executada, é realizar um cálculo da diferença entre o modelo candidato e os modelos que possuam um menor número de k e coerência similar. Entende-se como uma coerência similar valores com uma diferença inferior a 2% da maior coerência encontrada. Este processo é executado, pois, após testes experimentais, observou-se que modelos com um maior $k$ e coerência similar possuíam uma menor representatividade na classificação dos documentos, de forma que diversos tópicos eram "desprezados" no processo da classificação do corpus (mais detalhes sobre estes testes na seção dos experimentos e resultados 4.

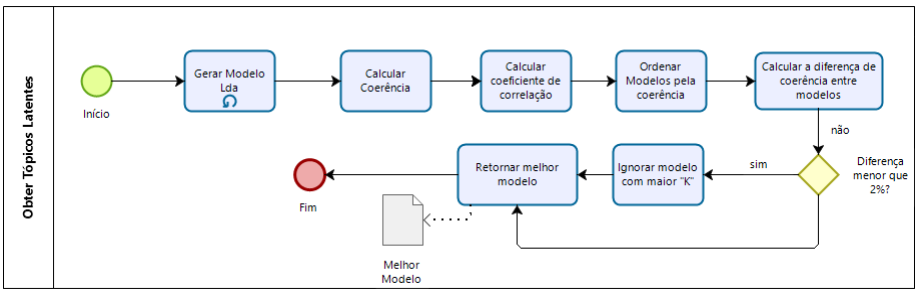



Fig. 1: Obtenção dos tópicos latentes

**Algoritmo 1:** Método proposto para a seleção dinâmica do melhor modelo LDA

```
   input  : janela_deslizante: Inteiro que determinará o tamanho da janela a ser utilizada para o cálculo da
            correlação
 1 Function obter_melhor_modelo(janela_deslizante):
 2     while correlacao > 0 do
 3         num_topicos +=1 ;
 4         modelo = treinar_modelo_lda(num_topicos)
 5         if (num_topicos % janela_deslizante)==0 then
 6             correlacao = calcular_correlacao(lista_modelos[janela_deslizante:]);
 7     melhor_modelo = selecao_modelo(lista_modelos)
 8     return melhor_modelo

   input  : lista_modelos: lista de objetos contendo os modelos, coerência e número de tópicos
 9 Function selecao_modelo(lista_modelos):
10     maior_corencia = max(lista_modelos.coerencia);
11     lista_corencia_similar = list(filter(lambda x: (maior_corencia - x)<=0.02, lista_modelos.coerencia))
12     melhor_modelo = min(lista_corencia_similar.num_topicos)
13     return melhor_modelo
```

Por fim, após a obtenção do modelo mais apropriado, todos os documentos do corpus são classificados com este modelo a fim de que cada vídeo da base esteja incluído em um ou mais tópicos. Em outras palavras, cada transcrição será inserida no modelo a fim do modelo inferir quais tópicos melhor descrevem cada documento.

## 4. EXPERIMENTOS E RESULTADOS

Para a avaliação do método proposto foram realizados experimentos com o objetivo de: (i) treinar diversos modelos de modelagem de tópicos para encontrar o melhor número de tópicos a partir de sua coerência; (ii) avaliar a representatividade dos tópicos do modelo selecionado na classificação dos documentos do corpus; e, (iii) avaliar o critério de seleção dinâmica de modelos proposto.

Foram coletados 250 vídeos de 5 canais diferentes, descritos na tabela I. Estes canais foram escolhidos por abordarem domínios distintos entre si (desde um canal sobre assuntos militares, até um canal sobre finanças) e por possuírem mais de 100 mil inscritos e mais de 100 vídeos postados. Todos os vídeos foram coletados no dia 10 de junho de 2020 e suas transcrições foram geradas através do algoritmo de ASR disponibilizado pelo próprio Youtube. Como critério de seleção, foram recuperados os 50 vídeos mais relevantes de cada canal[3]. O corpus resultante está disponível para consulta e download[4]. Cabe ressaltar, por fim, que as transcrições geradas possuem mais de 470 mil tokens.

Table I: Lista de canais em português (brasileiro) extraídos.

| Nome | Categoria | Número de vídeos |
|---|---|---|
| Hoje no Mundo Militar | Militar | 50 |
| Nerdologia | Ciências e cultura pop | 50 |
| Meteoro Brasil | Política e cultura pop | 50 |
| Filipe Deschamps | Programação e tecnologia | 50 |
| O Primo Rico | Finanças | 50 |

[3]A relevância dos vídeos é uma métrica desenvolvida pela própria plataforma com objetivo de recomendar os melhores vídeos para o usuário.
[4]https://www.ppgia.pucpr.br/~paraiso/Projects/YouGraph/

Após realizar o pré-processamento dos dados, realizamos a extração dos tópicos latentes presentes nas transcrições. Seguindo o método de critério de parada, foram gerados 58 modelos de modelagem de tópicos utilizando o algoritmo LDA com uma janela deslizante de 10 modelos (coeficiente de correlação = -0.13). A diferença entre cada modelo é o número de tópicos $k$ gerados. A cada modelo gerado, o número k era incrementado em 1, começando no primeiro modelo com um $k = 1$ e o último modelo com um $k = 58$. Ao final deste processo é calculada a coerência destes modelos para determinar qual seria o melhor modelo (lembrando que quanto maior a coerência melhor o modelo). A figura 2 apresenta o ganho de coerência entre os modelos. O modelo com a maior coerência, é o modelo com 40 tópicos e uma coerência de 44%.

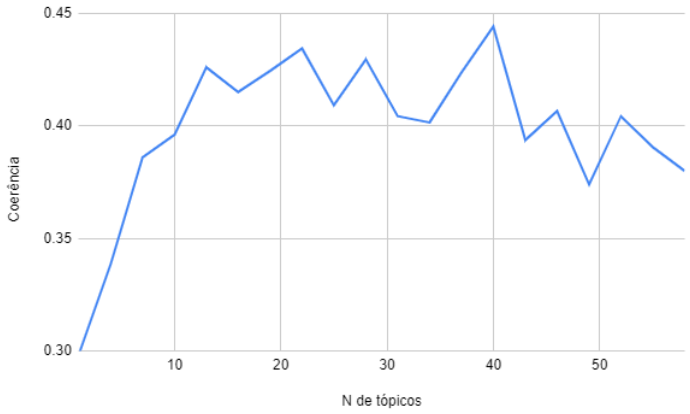
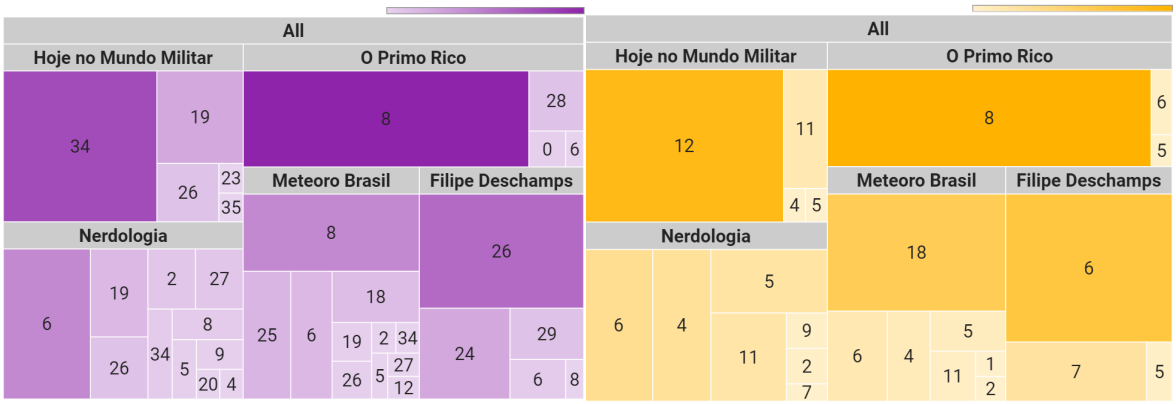



Fig. 2: Distribuição dos tópicos criados pelos canais

Na figura 3a, é possível verificar a distribuição dos tópicos criados pelos canais presentes utilizando este modelo. Nesta figura os números dentro dos quadros representam os ids de cada tópico e quanto mais forte a sua cor e maior o seu tamanho, maior é a quantidade vídeos classificados dentro deste tópico. Na tabela II é apresentada a composição dos principais tópicos do modelo a partir de seu id. Desta forma é possível compreender os significados dos tópicos a partir das palavras chaves os compõe. Analisando a figura é possível notar que o canal "Hoje no Mundo Militar" é majoritariamente composto pelo tópico de id 34 (64% dos vídeos), já o canal "Primo Rico", pelo tópico 8 (82%), por exemplo.




(a) Modelo com 40 tópicos

(b) Modelo com 19 tópicos

Fig. 3: Distribuição dos tópicos criados pelos canais

Seguindo o método proposto (seleção de um modelo com um menor k e uma diferença na coerência <2%), foi selecionado o modelo com 19 tópicos e uma coerência de 42%. Na figura 3b é possível visualizar a composição dos canais pelos tópicos classificados em cada vídeo. Ao confrontarmos a figura 3b com a 3a, é possível observar que ambos os modelos possuem um comportamento similar, mostrando um predomínio de determinados tópicos em cada canal. Isto também se comprova ao compararmos a composição das palavras chave dos tópicos dos modelos com k=19 e k=40 (veja as Tabelas III e II).

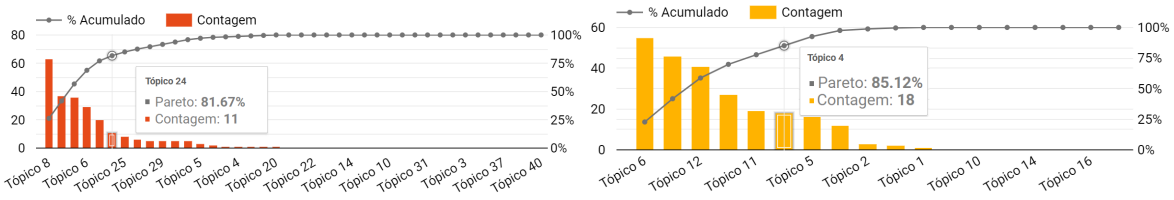
Table II: Palavras componentes dos tópicos latentes utilizando modelo com $k = 40$.

| **Tópicos** | **Palavras-chave** |
|---|---|
| topico_6 | gente - noto - filmar - médio - pessoa - gráfico - exemplo - tipo - coisa - casar - vidar - bom maneiro - cursar -diferente - pai - mundo - mear- novo - ação |
| topico_8 | gente - pessoa - coisa - dinheiro - bom - negócio - grande - dia - caro - mundo - crise momento - tempo - horar - empresar - primeiro - casar - seguinte - ano |
| topico_26 | vídeo - coisa - pessoa - canal - partir - tempo - projeto -formar - mundo - primeiro - importante novo - dia - sistema - tecnologia - exemplo - semana - repositório - nível - tipo |
| topico_34 | missão - guerra - militar - primeiro - mundo - estados_unidos - caçar - capaz grande - armar - unidade - dia - combater - canal - novo - avião - voar - ano - marinho |

Table III: Palavras componentes dos tópicos latentes utilizando modelo com $k = 19$.

| **Tópicos** | **Palavras-chave** |
|---|---|
| topico_6 | gente - pessoa - coisa - vídeo - filmar - mundo - tempo - exemplo - bom - canal - vidar - novo - primeiro - tipo - médio - melhor - formar - partir - legal - dia |
| topico_8 | gente - pessoa - dinheiro - coisa - negócio - ações - bom - empresar - grande - dia - caro - carteiro - vídeo - crise - ano - mercado - né - bolsa - mundo - momento |
| topico_7 | gente - jogar - código - coisa - formar - jogador - função - vídeo - pontar - novo - informação - servidor - simples - objeto - módulo - cliente - importante - variável - evento - nome |
| topico_12 | missão - militar - mundo - estados_unidos - caçar - canal - unidade - dia - capaz - grande - primeiro - novo - avião - voar - combater - aeronave - marinho - velocidade - vídeo - alvo |
| topico_18 | brasil - presidente - bolsonaro - governar - político - dia - país - brasileiro - momento - crime - pandemia - reunião - maio - público - ministrar - vidar - aplauso - músico - trabalhar - mulher |

Na figura 4a é apresentada a distribuição dos tópicos em relação ao número de vídeos classificados em cada tópico no modelo k=40. Como é possível notar os 6 tópicos mais frequentes (15% dos tópicos) representam 81% da base de dados. Além disso, é possível verificar que diversos tópicos sequer foram classificados em algum vídeo. A presença de tópicos sem classificação pode ser justificada pelo fato de que a obtenção dos tópicos latentes é feita separadamente da classificação dos vídeos. Como o algoritmo classifica cada transcrição retornando uma lista de probabilidades para cada vídeo, estes tópicos possuem uma probabilidade mínima de aparecerem em cada vídeo. Já na figura 4b, podemos observar a distribuição dos tópicos no modelo k=19 (que possui uma coerência de 42%). Neste modelo os 6 tópicos (31% dos tópicos) mais frequentes, representam 85% da base de dados.




(a) Distribuição com modelo de 40 tópicos

(b) Distribuição com modelo de 19 tópicos

Fig. 4: Distribuição de Pareto dos modelos

## 5. CONCLUSÃO

Este artigo apresentou um estudo sobre a utilização da modelagem de tópicos a partir do algoritmo LDA em transcrições de áudio oriundos de vídeos do Youtube em português brasileiro. Foi apresentado também, uma abordagem para permitir a seleção dinâmica do número ideal de tópicos que compõe um dado corpus. Utilizando o método desenvolvido foi possível correlacionar diferentes transcrições de conteúdos (tópicos) a partir de seus canais de origem e entre canais. O critério de corte a partir da coerência também mostrou que é possível melhorar as distribuições dos documentos entre os tópicos, selecionando um modelo com um $k$ menor. Isto permite uma melhor generalização do modelo, mantendo a sua estrutura e composição das palavras chaves dos tópicos.

Em estudos futuros, pretende-se analisar o comportamento do método em um corpus com mais transcrições, bem como aplicar um *pos-tagger* com intuito de avaliar se a remoção de diferentes *part-of-speech* (como por exemplo, verbos e advérbios) podem melhorar os resultados através do aumento da semântica dos dados e consequentemente da coerência dos modelos. Por fim, pretende-se também, analisar a presença de tópicos emergentes em relação ao tempo. Isto é interessante pois possibilitaria visualizar se os autores tendem a seguir tendências que surgem em um determinado tema.

## REFERENCES


Alexe, B., Hernandez, M. A., Hildrum, K. W., Krishnamurthy, R., Koutrika, G., Nagarajan, M., Roitman, H., Shmueli-Scheuer, M., Stanoi, I. R., Venkatramani, C., et al. Surfacing time-critical insights from social media. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. pp. 657–660, 2012.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research* 3 (Jan): 993–1022, 2003.

de Souza, M. and Souza, R. R. Modelagem de tópicos. *Múltiplos Olhares em Ciência da Informação* 9 (2), 2019.

Gausby, A. Attention spans. *Consumer Insights, Microsoft Canada*, 2015.

Hagen, L. Content analysis of e-petitions with topic modeling: How to train and evaluate lda models? *Information Processing & Management* 54 (6): 1292–1307, 2018.

He, Q., Chen, B., and Argawal, D. Building the linkedin knowledge graph. *LinkedIn*, 2016.

Kaushik, L., Sangwan, A., and Hansen, J. H. Automatic sentiment extraction from youtube videos. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, pp. 239–244, 2013.

Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus* 5 (1): 1608, 2016.

Misra, H., Yvon, F., Jose, J. M., and Cappe, O. Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM conference on Information and knowledge management*. pp. 1553–1556, 2009.

Munaro, A. C., Barcelos, R. H., Maffezzolli, E. C. F., Rodrigues, J. P. S., and Paraiso, E. C. The drivers of video popularity on youtube: An empirical investigation. In *Advances in Digital Marketing and eCommerce*. Springer, pp. 70–79, 2020.

Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., and Taylor, J. Industry-scale knowledge graphs: Lessons and challenges. *Queue* 17 (2): 48–75, 2019.

O'callaghan, D., Greene, D., Carthy, J., and Cunningham, P. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications* 42 (13): 5645–5657, 2015.

Patel, A. B., Birla, M., and Nair, U. Addressing big data problem using hadoop and map reduce. In *2012 Nirma University International Conference on Engineering (NUiCONE)*. IEEE, pp. 1–5, 2012.

Rangaswamy, S., Ghosh, S., Jha, S., and Ramalingam, S. Metadata extraction and classification of youtube videos using sentiment analysis. In *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*. IEEE, pp. 1–2, 2016.

Syed, S. and Spruit, M. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International conference on data science and advanced analytics (DSAA)*. IEEE, pp. 165–174, 2017.

Wöllmer, M., Weninger, F., Knaup, T., Schuller, B., Sun, C., Sagae, K., and Morency, L.-P. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems* 28 (3): 46–53, 2013.

Youtube. Youtube in numbers, 2020.