UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ian Macedo Maiwald Santos

Explorando a Modelagem de Tópicos em Textos Curtos de Mídias Sociais: uma Análise Comparativa de Algoritmos

Florianópolis
2023

Ian Macedo Maiwald Santos

Explorando a Modelagem de Tópicos em Textos Curtos de Mídias Sociais: uma Análise Comparativa de Algoritmos

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Ciência da Computação.
Orientadora: Profa. Luciana de Oliveira Rech, Dra.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Santos, Ian Macedo Maiwald
Explorando a Modelagem de Tópicos em Textos Curtos de Mídias Sociais : uma Análise Comparativa de Algoritmos / Ian Macedo Maiwald Santos ; orientadora, Luciana de Oliveira Rech, 2023.
148 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciência da Computação. 2. Modelagem de tópicos. 3. Textos curtos. 4. Mídias sociais. I. Rech, Luciana de Oliveira. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Ian Macedo Maiwald Santos

Explorando a Modelagem de Tópicos em Textos Curtos de Mídias Sociais: uma Análise Comparativa de Algoritmos

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Ricardo Alexandre Reinaldo de Moraes, Dr.
Instituição UFSC

Prof. Mauro Roisenberg, Dr.
Instituição UFSC

Prof. Gustavo Medeiros de Araújo, Dr.
Instituição UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Ciência da Computação.

Documento assinado digitalmente
PATRICIA DELLA MEA PLENTZ
Data: 23/08/2023 16:19:48-0300
CPF: ***.243.700-**
Verifique as assinaturas em https://v.ufsc.br

Coordenação do Programa de Pós-Graduação

Documento assinado digitalmente
Luciana de Oliveira Rech
Data: 16/08/2023 07:29:47-0300
CPF: ***.714.910-**
Verifique as assinaturas em https://v.ufsc.br

Profa. Luciana de Oliveira Rech, Dra.
Orientadora

Florianópolis, 2023.

Este trabalho é dedicado à minha amada família e meus queridos amigos.

AGRADECIMENTOS

Ao fim desta jornada, olho para trás e só tenho a agradecer aqueles que estiveram comigo. Sem o conforto da companhia de todos vocês, eu não chegaria a lugar algum. Recordarei para sempre os bons e maus momentos, e agora posso ver como cada um deles contribuiu para que este trabalho pudesse chegar à sua conclusão.

Primeiramente agradeço à minha família, que sempre me incentivou e me mostrou a importância do conhecimento. Agradeço pela confiança em minhas capacidades e por todo o investimento em minha jornada até aqui. Muito obrigado mãe, pai, tias e tios, primas e primos, e tantos outros familiares que torceram por mim!

Agradeço meus amigos que, apesar da distância, me ajudaram e incentivaram a todo momento. Desde o ano de 2015, quando comecei minha jornada acadêmica, me sinto extremamente privilegiado de poder contar com o apoio de tantos companheiros talentosos na minha área de pesquisa. Os momentos felizes e de perrengue, as risadas, o suor e o sangue que derramamos juntos — e uns pelos outros — estão eternizados no meu baú de tesouros.

Por fim, não posso deixar de agradecer aos maravilhosos professores que me acompanharam por todo esse processo, em especial aos membros da banca e à minha orientadora. A experiência e sabedoria deles me amparou nos diversos momentos dessa montanha-russa de emoções, e foram essenciais na construção do conhecimento descrito por esta dissertação.

RESUMO

Diariamente, muitos textos são publicados em massa nas mídias sociais. A compreensão dos temas e padrões nessas discussões é crucial para contextos como grandes eventos esportivos, desastres naturais ou eleições. Para esses cenários, a modelagem de tópicos é uma técnica de Processamento de Linguagem Natural que identifica os tópicos mais relevantes em uma coleção de textos. No entanto, para textos curtos, como os das mídias sociais, é muito comum o uso de técnicas inadequadas, que obtêm resultados com prejuízos devido à natureza reduzida desses textos. À vista disso, este trabalho avaliou o desempenho de algoritmos de modelagem de tópicos de quatro categorias: Tradicionais, baseados em DMM, baseados em autoagregação e baseados em coocorrência global. Para a análise, foram utilizados textos reais, publicados por usuários de mídias sociais. Os desempenhos dos modelos foram avaliados com o auxílio de métricas de qualidade, identificadas na bibliografia. Para isso, foram mensurados critérios de qualidade dos tópicos gerados pelos modelos, tais como a capacidade de generalização dos modelos, a semelhança dos tópicos, e a coerência das palavras que formam cada tópico. Os resultados mostraram que a categoria de algoritmos Tradicionais tiveram as piores capacidades de generalização, e também os menores desempenhos de distância, divergência e coerência nos tópicos. Com isso, fica evidente a incapacidade desses algoritmos em se adequarem aos textos curtos. Em alguns casos específicos, os modelos Tradicionais geraram tópicos com alguma coesão entre as palavras, porém ficou evidente o prejuízo nos produtos. Além disso, observou-se que cada técnica de modelagem possui um uso mais adequado conforme o propósito da análise que se busca realizar no corpus. Para uma análise voltada às nuances de um tema principal, os algoritmos baseados em DMM mostraram o maior potencial. Com distância e divergência mais baixas que a média, e coerências razoáveis, essa categoria mostrou resultados alinhados a uma análise de subtemas. Por outro lado, para análises focadas na exploração e identificação de temas distintos no conjunto de dados, o algoritmo BTM se destacou sozinho mostrando valores acima da média na Distância, Divergência, e Coerências. Ele gerou tópicos bem separados, que forneceram ideias mais claras e úteis. Além desses dois polos, as categorias de algoritmos baseados em Coocorrência global e de Autoagregação se mostraram mais equilibradas, sem apresentarem tendências muito evidentes para uma Análise de Subtemas, ou para uma Análise Exploratória.

Palavras-chave: Modelagem de tópicos. Textos curtos. Mídias sociais.

ABSTRACT

Daily, many texts are mass published on social media platforms. Understanding the themes and patterns in these discussions is crucial for contexts such as sporting events, natural disasters or elections. In these scenarios, topic modeling comes as a Natural Language Processing technique that identifies the most relevant topics in a collection of texts. However, for short texts, such as those on social media, inappropriate techniques are very commonly used, which presents results with losses due to the reduced nature of these texts. In view of this, this study evaluated the performance of topic modeling algorithms from four categories: Traditional, DMM-based, Self-Aggregation-based and Global Co-occurrence-based. For the analysis, real texts published by social media users were used. The performance of the models were evaluated with the help of quality metrics, identified in the literature. For this, quality criteria of the topics generated by the models were measured, such as the generalization capacity of the models, the similarity of the topics, and the coherence of the words that form each topic. The results showed that the category of Traditional algorithms had the worst generalization capabilities, and also the lowest performances of distance, divergence and coherence in the topics. With this, it is evident the inability of these algorithms to adapt to short texts. In some specific cases, the Traditional models generated topics with some cohesion between the words, but the loss in the final products was evident. In addition, it was observed that each modeling technique has a more appropriate use according to the purpose of the analysis that is sought in the corpus. For an analysis focused on the nuances of a main theme, DMM-based algorithms showed the greatest potential. With lower than average distance and divergence, and reasonable coherences, this category showed results aligned with an analysis of sub-themes. On the other hand, for analyses focused on exploring and identifying distinct themes in the dataset, the BTM algorithm stood out alone showing above average values in Distance, Divergence, and Coherences. It generated well-separated topics, which provided clearer and more useful insights. Besides these two poles, the categories of algorithms based on Global Cooccurrence and Self-Aggregation were more balanced, without showing much evident tendencies for a Subtheme Analysis, or for an Exploratory Analysis.

Keywords: Topic modeling. Short texts. Social media.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BTM	*Biterm Topic Modeling*
CRF	*Conditional Random Fields*
CRFTM	*Conditional Random Field Regularized Topic Model*
DMM	*Dirichlet Multinomial Mixture*
EM	*Expectation-Maximization*
EMAD	*Embedding-Based Minimum Average Distance*
GMM	*Gaussian Mixture Model*
GPU	*Generalized Polya urn*
GPU-PDMM	*Generalized Polya urn Poisson* DMM
LapDMM	*Laplacian* DMM
LDA	*Latent Dirichlet Allocation*
LSA	*Latent Semantic Analysis*
MIGA	*Meta-Info Guided Aggregation*
MLP	Perceptron multicamada
MTM	*Multiterm Topic Model*
NBTMWE	*Noise Biterm Topic Model with Word Embeddings*
NMF	*Non-negative Matrix Factorization*
NQTM	*Negative sampling and Quantization Topic Model*
PCA	Análise de Componentes Principais
PLN	Processamento de Linguagem Natural
PLSA	*Probabilistic Latent Semantic Analysis*
PTM	*Pseudo-Document-Based Topic Modeling*
SATM	*Self-Aggregation Based Topic Model*
SVD	*Singular Value Decomposition*
TF-IDF	*Term Frequency–Inverse Document Frequency*
WNTM	*Word Network Topic Model*

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A popularização das mídias sociais possibilita que as pessoas participem de discussões online, permitindo que qualquer um contribua e opine sobre diversos temas. Com essa grande participação, grandes fluxos de dados são gerados e as plataformas online são inundadas, majoritariamente, com mensagens na forma de textos curtos (PATHAK; PANDEY; RAUTARAY, 2019).

O uso das mídias sociais tem crescido exponencialmente, com um aumento considerável no número de usuários ativos. Segundo a (STATISTA, 2020), em 2020 houve um registro de 3,6 bilhões de usuários ativos, com estimativas que em 2025 esse número alcance a marca de 4,41 bilhões. Como resultado, uma enorme quantidade de dados é gerada, incluindo mensagens em perfis pessoais, comentários em blogs e discussões em fóruns online.

Compreender os eventos em andamento nas mídias sociais é essencial para alcançar um ponto de vista sobre as tendências, que podem influenciar atitudes e comportamentos individuais (WILLIAMS *et al.*, 2015). Entretanto, em discussões online com muitos participantes, alguns assuntos são mais evidentes, enquanto outras tendências são mais discretas e passam despercebidas (MCQUILLAN *et al.*, 2020).

A pandemia de COVID-19 ressaltou a importância da utilização de grandes conjuntos de dados para fins científicos, demonstrando como isso pode ajudar a salvar vidas (NATURE, 2021). Outras circunstâncias em que isso se aplica são em discussões relacionadas a desastres naturais, eleições e eventos esportivos.

Outro ponto a se destacar é que, diante do enorme volume de dados gerados pelas mídias sociais, a realização de análises manuais torna-se inviável. Contudo, a necessidade de compreender o que as pessoas discutem nessas plataformas torna-se mais emergente.

Nesse contexto surge a modelagem de tópicos, uma técnica de mineração de texto que identifica estruturas semânticas em um corpus de documentos. O Latent Dirichlet Allocation (LDA) de Blei, Ng e Jordan (2003), por exemplo, é uma solução não supervisionada bastante difundida para realizar essa modelagem, que pode ser utilizada em diversas áreas, como ciência política (GREENE; CROSS, 2015; ALASHRI *et al.*, 2016), ciência médica (ZHANG, Y. *et al.*, 2017), e para identificar redes de desinformação (MCQUILLAN *et al.*, 2020). Mais precisamente, trata-se de um modelo estatístico que revela os conjuntos de palavras semanticamente relacionadas mais proeminentes em uma coleção de textos (BLEI; LAFFERTY, 2006).

No entanto, a aplicação efetiva das soluções de modelagem de tópicos em mídias sociais enfrenta diversos desafios. Um aspecto predominante nos textos das mídias sociais é seu tamanho curto, tornando a obtenção de resultados eficientes um

desafio quando se aplicam às abordagens tradicionais de modelagem de tópicos. Isso se deve principalmente à esparsidade dos dados, já que os textos curtos possuem um contexto naturalmente limitado (WU, X.; LI, Chunping, 2019).

O LDA, por exemplo, é uma técnica amplamente utilizada. Entretanto, a aplicação dessa solução em documentos curtos pode não ser adequada. Abordagens tradicionais consideram que as palavras em cada documento possuem uma relação entre si, e a partir desta característica, identifica no corpus a distribuição dos tópicos e as palavras de cada um deles. Portanto, tornando-se inadequado quando se trata de documentos com tamanho reduzido (CHENG *et al.*, 2014).

Outros modelos como o Biterm Topic Modeling (BTM) surgem como alternativas para contornar esse problema. Esses modelos mapeiam padrões globais de co-ocorrência de palavras no corpus, o que permite buscar relações semânticas entre palavras além do nível de documento (RASHID; SHAH; IRTAZA, 2019). Além disso, esses modelos conseguem funcionar independentemente, sem a necessidade de um vocabulário externo. (QIANG *et al.*, 2020) listam três categorias principais de modelagem de tópicos para textos curtos: modelos baseados em Dirichlet Multinomial Mixture (YIN; WANG, 2014; LI, Chenliang *et al.*, 2017), modelos baseados em co-ocorrência global (CHENG *et al.*, 2014; HADI; FARD, 2020), ou métodos de autoagregação (ZHAO, H. *et al.*, 2019; GAO, W. *et al.*, 2019).

Em um processo de modelagem de tópicos, é fundamental considerar as diversas características que o conteúdo do corpus possui. Em se tratando de mídias sociais, além dos textos serem majoritariamente curtos, também destaca-se que são informais (PATHAK; PANDEY; RAUTARAY, 2019) e, muitas vezes, acompanhados de SPAM realizado por agentes automatizados (HAUPT *et al.*, 2021). Compreender esses aspectos é fundamental para obter resultados mais precisos e confiáveis na modelagem de tópicos de mídias sociais.

Apesar da existência de soluções específicas para esse tipo de dado, modelos tradicionais como LDA ainda são amplamente utilizados na prática. Além disso, ressalta-se que a aplicação dessas técnicas em outros idiomas além do inglês é pouco explorada na literatura científica.

Assim sendo, esta pesquisa visa responder a seguinte questão: com base na língua portuguesa,**quais técnicas de modelagem de tópicos mostram-se adequadas e eficientes para identificar tópicos de interesse em textos provenientes de mídias sociais?**

Para responder à pergunta, serão aplicados diferentes algoritmos de modelagem de tópicos em um corpus de tuítes. Cada processo será avaliado considerando a perplexidade, distância de tópicos, divergência-KL simétrica, e coerência de tópicos. Na sequência, as técnicas serão comparadas entre si para destacar as suas vantagens e limitações.

Os resultados obtidos por uma modelagem de tópicos de textos de mídias sociais podem revelar o que as pessoas discutem em plataformas online a respeito de um determinado tema. No caso da pandemia de COVID-19, por exemplo, a análise dos tópicos pode ajudar a identificar as principais preocupações, dúvidas e opiniões da população em relação à doença.

Tendo isso em vista, este estudo avalia as técnicas de modelagem de tópicos mais adequadas para identificar tópicos latentes em conteúdos textuais de mídias sociais. O corpus de textos utilizado para essa análise é composto por tuítes em português sobre a pandemia da COVID-19.

1.2 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos.

1.2.1 Objetivo Geral

Comparar o desempenho de abordagens tradicionais de modelagem de tópicos em relação a outras categorias mais especializadas, na aplicação em uma coleção de textos curtos provenientes de mídias sociais, identificando como cada categoria de algoritmo se comporta em um cenário real.

1.2.2 Objetivos Específicos

1. Identificar, através de uma pesquisa bibliográfica, as técnicas de modelagem de tópicos para textos curtos;
2. Definir um fluxo operacional para a modelagem de tópicos de coleções de textos de mídias sociais;
3. Montar um conjunto de textos proveniente de discussões reais em uma plataforma de mídias sociais;
4. Definir as métricas adequadas para avaliação dos algoritmos de modelagem de tópicos;
5. Implementar os algoritmos de modelagem de tópicos selecionados para realização de uma análise comparativa de seus respectivos desempenhos.

1.3 CONTRIBUIÇÕES

As principais contribuições deste trabalho são:

1. Uma seleção de técnicas de modelagem de tópicos para serem aplicadas em textos curtos, e de métricas para avaliar o desempenho desses algoritmos;

2. Um fluxo metodológico para modelagem de tópicos em textos originados em mídias sociais;
3. A implementação dos algoritmos selecionados, incluindo otimizações de paralelização e uso de matrizes esparsas;
4. Uma análise comparativa dos desempenhos dos algoritmos quando aplicados a textos curtos reais, usando as métricas selecionadas.

1.4 ORGANIZAÇÃO DO TEXTO

Este capítulo descreve, principalmente, as motivações, objetivos e justificativa para a realização deste trabalho. Os demais capítulos deste documento estão organizados da seguinte forma:

- **Capítulo 2 - Fundamentação teórica**: para sustentar o trabalho proposto, são descritos os conceitos sobre modelagem de tópicos, vetores e modelos, processamento de linguagem natural (PLN), e mídias sociais;
- **Capítulo 3 - Métricas de avaliação**: neste capítulo são listadas as métricas de avaliação usadas para medir o desempenho dos algoritmos selecionados. É apresentada uma descrição sobre como a métrica atua e do significado que possuem os resultados obtidos pela avaliação;
- **Capítulo 4 - Trabalhos relacionados**: este capítulo lista e descreve os algoritmos selecionados para serem avaliados neste trabalho. Os algoritmos foram classificados entre 4 categorias comumente adotadas pela comunidade científica. Cada algoritmo possui uma subseção própria onde seu funcionamento é detalhado. O capítulo conta também com uma seção sobre Características gerais sobre essas técnicas, que possuem observações pertinentes sobre elas;
- **Capítulo 5 - Procedimentos de análise de algoritmos**: descreve o procedimento metodológico adotado para este trabalho, abrangendo os materiais e métodos envolvidos. Desde a coleta dos dados até o processamento aplicado nos textos;
- **Capítulo 6 - Resultados e Discussão**: avalia e discute os resultados observados na avaliação das métricas, conforme expostos no Apêndice A. Os valores são comparados mais amplamente, considerando todo o espectro de técnicas selecionadas neste trabalho. Também são apresentadas as considerações gerais a respeito dos resultados, com os algoritmos que mais se destacaram nas avaliações;
- **Capítulo 7 - Conclusões e Trabalhos futuros**: para concluir esta dissertação, o capítulo traz uma revisão das motivações e objetivos, seguido de

uma visão geral do que foi apresentado. Também descreve as contribuições dessa dissertação para a comunidade científica e, por fim, traz sugestões de temas que ainda podem ser explorados nas análises comparativas sobre modelagem de tópicos.

- **Apêndice A**: apresenta um relatório com os resultados obtidos por meio de gráficos de colunas agrupadas e tabelas, proporcionando uma visão abrangente e compreensível do desempenho dos algoritmos avaliados com base nas métricas selecionadas.

2 FUNDAMENTAÇÃO TEÓRICA

Para apoiar a compreensão desta dissertação, neste capítulo serão expostos: conceitos básicos de modelagem de tópicos, incluindo vocabulário comumente usado e como vetores são utilizados; a relação das técnicas de modelagem, o PLN, e o *clustering* do aprendizado não supervisionado; e a relevância das mídias sociais como ambiente de grandes discussões.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

O PLN é uma ramificação da inteligência artificial com foco em permitir que computadores possam manipular, interpretar, e compreender a linguagem humana (CHOWDHARY, 2020). A associação entre a ciência da computação e linguística computacional são a base para preencher as lacunas entre comunicação humana e compreensão dos computadores. Com PLN é possível analisar textos. Deles, pode-se extrair informações sobre pessoas, lugares, e eventos para compreender, por exemplo, os sentimentos dos usuários de uma mídia social e suas conversas (COPPERSMITH *et al.*, 2018).

O PLN busca o entendimento da linguagem natural para compreender os significados latentes em um corpo de texto. É possível categorizar, arquivar, e analisar textos; viabilizando uma tomada de decisões baseada nos resultados (INDURKHYA; DAMERAU, 2010). A partir de aplicações de PLN, dados textuais não-estruturados podem ser compreendidos por meio de percepções obtidas na informação extraída.

A história do PLN data desde que Turing (1950) apresenta o que hoje é chamado de teste de Turing. Desde então, os avanços na área se beneficiaram com o crescente interesse na comunicação entre homem-computador. O progresso constante da capacidade dos algoritmos é possível devido à disponibilidade de grandes volumes de dados e melhorias computacionais.

A linguagem de máquina é nativa dos computadores, e complexa para o entendimento humano. Nesse caso, a comunicação não acontece por palavras, mas por milhões de zeros e uns, resultando em ações lógicas (HIRSCHBERG; MANNING, 2015). O PLN é um intermediário na comunicação entre humano e computador, interpretando a linguagem nativa para realizar tarefas (CHOWDHARY, 2020). Com isso, máquinas podem compreender textos e falas, enquanto determinam as partes importantes de cada para medir as ideias envolvidas no processo.

Uma máquina moderna pode superar um humano em análise de dados baseados em linguagem; consistentemente, sem fadiga, e imparcial (COPPERSMITH *et al.*, 2018). Com a abundância de dados não estruturados gerados diariamente, desde registros médicos até publicações em mídias sociais, a automação da análise de texto mostra-se essencial.

As diversas variações dos dados baseados em texto e voz fomentaram uma ampla gama de soluções. Devido a isso, o PLN abrange muitas técnicas para interpretar a linguagem humana (NADKARNI; OHNO-MACHADO; CHAPMAN, 2011), que incluem: métodos estatísticos, aprendizado de máquinas, e abordagens baseadas em regras e algorítmicas. Tarefas básicas do PLN incluem tokenização e *parsing*, lematização/stemização, rotulagem de discurso, detecção de idioma e identificação de relacionamentos semânticos (DENG; LIU, Y., 2018).

Em suma, o PLN divide a linguagem em partes menores, entende a relação entre os pedaços e explora como as partes, juntas, criam significado. Tarefas assim são comumente empregadas em níveis mais complexos de PLN, como:

- **categorização de conteúdo**: produzir um resumo do documento, incluindo pesquisa e indexação, alerta de conteúdo e detecção de cópias (CHEN, P.-H. *et al.*, 2018);
- **descoberta e modelagem de tópicos**: capturar os temas e significados latentes em coleções de textos (HAGEN *et al.*, 2015);
- **análise de corpus**: entender a estrutura de um corpus e seus documentos por meio de estatísticas para tarefas como amostragem, preparação de dados para outros modelos e abordagens de modelagem (SIMPSON *et al.*, 2018);
- **extração de contexto**: extrair automaticamente informação estruturada de fontes de texto (TIXIER *et al.*, 2016);
- **análise de sentimento**: minerar opiniões e identificar o sentimento médio em grandes quantidades de texto (RAJPUT, 2020);
- **conversão fala-texto e texto-fala**: converter comandos de voz em texto escrito, e vice-versa (SANTRA *et al.*, 2018);
- **sumarização de documento**: gerar sinopses automáticas de grandes corpos de texto e detectar idiomas presentes em corpus multilíngues (AWASTHI *et al.*, 2021);
- **tradução**: traduzir automaticamente texto ou fala de um idioma para outro (ZONG; HONG, C., 2018).

Em todos esses casos, o processo consiste em combinar linguística e algoritmos para transformar a linguagem bruta em um produto de maior valor. Para isso, o aprendizado não supervisionado pode ser um grande aliado, especialmente quando se trata de modelagem de tópicos.

2.2 MODELAGEM DE TÓPICOS

Em uma análise de dados, um dos principais objetivos é determinar as características compartilhadas por unidades de informação. Na análise textual, isso pode se traduzir em determinar quais conceitos ou eventos são discutidos em um documento. Um humano consegue entender claramente um texto apenas com a leitura. Porém, para um programa, não é uma tarefa trivial extrair um contexto ou assunto da sequência de palavras apenas como está escrita.

Para um programa ter essa capacidade, uma alternativa popular é a modelagem de tópicos. Esse é um método estatístico usado para extrair variáveis latentes de grandes conjuntos de dados (BLEI, 2012). É uma solução particularmente apropriada para uso com dados de textos, porém também é usado em outras mídias como imagens (ZHOU, Z.; ZHOU, J.; ZHANG, L., 2016), vídeo (HOSPEDALES; GONG; XIANG, 2012), sequências genéticas e bioquímicas (LIU, L. *et al.*, 2016), e até dados geoespaciais (JU *et al.*, 2016).

Esse método busca gerar tópicos concisos e pertinentes que se relacionam bem com conceitos humanos. Cada tópico é formado por um conjunto de palavras que possuem uma relação semântica entre si. Portanto, esta é uma das razões do uso intensivo de modelagem de tópicos em pesquisas atuais nas áreas relacionadas ao Processamento de Linguagem Natural (PLN). Os dados analisados podem variar em fontes e formas, além de serem comumente classificados por hierarquias. Tendo em vista que esse trabalho trata de texto, a Figura 1 ilustra a hierarquia de tais dados.

Figura 1 – Hierarquia dos dados textuais.

Este trabalho envolve modelagem de tópicos em texto, e, para tanto, discorre sobre as expressões usadas nesse contexto. Uma “palavra” ou “termo” representa a unidade fundamental de dados individuais, uma coleção de palavras forma uma “sentença”; um “documento” representa uma ou mais sentenças composta por *N* palavras; e um “corpus” representa um conjunto formado por *M* documentos, geralmente englobando todo o conjunto de dados.

Um “vocabulário” é constituído pela coleção de todas as palavras distintas de um corpus; e um “tópico” caracteriza a distribuição de probabilidade que abrange um determinado vocabulário.

Em termos de algoritmo, as palavras podem ser designadas por vetores unitários que abrangem as dimensões de um vocabulário indexado por $\{1, \ldots, V\}$. Assim, considera-se que o V-ésimo termo do vocabulário seria apontado como um vetor com V dimensões w tal que $w^v = 1$ e $w^v = 0$ para $v \neq u$. Isso significa que, um único componente que representa a posição da palavra no vocabulário desses vetores é igual a um, e todos os outros componentes são iguais a zero.

Seguindo este raciocínio, um documento pode ser considerado uma matriz unitária representada por $w = (w_1, w_2, ..., w_N)$. Logo, w_i representa a i-ésima palavra de uma sequência e N define o número de palavras no documento.

À vista disso, um corpus é representado por $D = (w_1, w_2, ..., w_M)$ ou $D = (d_1, d_2, ..., d_M)$, em que d_n é equivalente a w_n, e significa o n-ésimo documento do corpus enquanto M define o número total de documentos no corpus. Em sua forma mais simples, sem considerar hierarquias ou relacionamentos sequenciais, pode ser representado por $z = (z_1, z_2, ..., z_K)$ com z_j representando o j-ésimo tópico e K definindo o número de tópicos que abrangem o corpus (BLEI; NG; JORDAN, 2003).

Mimno *et al.* (2011) qualificam tópicos em três níveis: bom, intermediário e ruim. Em geral, um tópico é considerado “bom” se contiver palavras que podem ser agrupadas coerentemente como um único conceito. Para os tópicos marcados como “intermediário” ou “ruim”, os autores identificaram que estes apresentam um ou mais dos seguintes problemas:

- **encadeamento**: cada palavra possui alguma conexão a todas as outras palavras por meio de uma cadeia de pares de palavras, mas nem todos os pares de palavras do tópico fazem sentido. Por exemplo, um tópico cujas três palavras mais importantes sejam “covid”, “doença” e “sarampo”. São dois conceitos distintos (covid e sarampo) encadeados pela palavra “doença”.
- **intruso**: dois ou mais conjuntos de palavras não relacionadas, unidos arbitrariamente sob um tópico. Um bom tópico com algumas palavras “intrusas”;
- **aleatório**: sem conexões claras e com sentido entre mais do que alguns pares de palavras; e
- **desequilíbrio**: as principais palavras estão logicamente conectadas umas às outras, mas o tópico combina termos muito gerais a específicos (por exemplo, “COVID-19” e “variante B.1.1.529”).

A depender do método utilizado, os tópicos são identificados por diferentes distribuições probabilísticas ou estocásticas. Em alguns casos também podem representar a distribuição sobre outros tópicos.

A demanda computacional dos algoritmos também é um fator a se considerar. Essa demanda está sujeita ao tamanho do corpus, o número de tópicos, o número de iterações e de outros fatores. Isso se dá, pois, quanto maior o corpus, ou seja, o número de documentos, maior será a demanda computacional. É preciso processar todos os documentos do corpus para estimar as distribuições de tópicos.

Da mesma forma, quanto maior o número de tópicos especificado para o modelo, maior será a demanda computacional, por ser necessário estimar as distribuições de palavras para cada tópico. O número de iterações no modelo também é uma influência, uma vez que cada iteração envolve atualizações dos parâmetros do modelo. Portanto, quanto maior o número de iterações, maior será o esforço.

Além desses fatores, as particularidades de cada algoritmo podem adicionar ainda mais à demanda, como a inclusão de hiperparâmetros adicionais, modelos de linguagem pré-treinados ou técnicas para auxílio de inferência, como o deep learning. Por isso é recomendado realizar testes e avaliar a demanda computacional para cada caso específico, especialmente em grandes conjuntos de dados.

2.2.1 Definição formal de um tópico

Formalmente, define-se um tópico como um vetor de probabilidade, onde cada elemento do vetor representa a probabilidade de uma palavra específica pertencer ao tópico (VAYANSKY; KUMAR, 2020). Suponha que temos um conjunto de documentos representado como uma matriz de termo-documento, onde cada linha representa um documento e cada coluna representa uma palavra do vocabulário. Podemos representar um tópico como um vetor θ de tamanho V, onde V é o tamanho do vocabulário e θ_i é a probabilidade da i-ésima palavra w_i pertencer ao tópico (CHURCHILL; SINGH, 2022; BLEI; NG; JORDAN, 2003).

Nesse caso, um tópico θ é uma distribuição de probabilidade, o que significa que seus elementos devem satisfazer as seguintes condições:

- $0 \leq \theta_i \leq 1$ para todo i (probabilidades não negativas)
- $\sum_{i=1}^{V} \theta_i = 1$ (a soma de todas as probabilidades é igual a 1)

Na prática, um algoritmo como o LDA, atribui uma distribuição de tópicos a cada documento no corpus. Cada documento é visto como uma mistura de tópicos, onde a proporção de cada tópico é determinada pelas probabilidades atribuídas aos tópicos no documento. Essas probabilidades são geralmente representadas como um vetor θ de tamanho K, onde K é o número de tópicos no modelo. Outras técnicas podem ter premissas diferentes, como o *Dirichlet Multinomial Mixture* (DMM), que presume que cada documento está atrelado a apenas uma distribuição de tópicos.

Por isso há um uso intensivo de modelagem de tópicos em pesquisas atuais nas áreas relacionadas ao PLN. Essas representações de distribuições de probabilidade

sobre as palavras do vocabulário são usadas para descrever a estrutura latente dos documentos e facilitar a análise de grandes conjuntos de documentos.

2.2.2 Objetivo de uma modelagem

Com os tópicos em mãos, a modelagem de tópicos permite um amplo espectro de análises que podem ser conduzidas em diferentes níveis de detalhamento e aplicabilidade (RANA; CHEAH; LETCHMUNAN, 2016; BARDE; BAINWAD, 2017). Alguns exemplos são as análises de nível Macro, Meso e Micro, ou Longitudinais. Contudo, tais análises estão sujeitas a abordagens exploratórias ou que envolvem subtemas.

Nesse caso, a análise exploratória é uma abordagem indutiva e sem compromisso com hipóteses pré-estabelecidas (ORDUN; PURUSHOTHAM; RAFF, 2020). Essa abordagem concentra-se na identificação de padrões, temas e correlações emergentes no corpus de texto sem qualquer suposição ou foco prévio. Este procedimento ajuda a descobrir novas relações e tendências, fornecendo uma visão abrangente do corpus e informando a formulação de futuras perguntas de pesquisa.

Já a análise de subtemas é uma abordagem orientada à hipótese, buscando extrair detalhes mais profundos e refinados a partir de um tema principal já reconhecido (MOHR; BOGDANOV, 2013). Ela se baseia em um entendimento preexistente do corpus e visa investigar mais minuciosamente as nuances em um tópico específico, permitindo a detecção de subtemas que possam estar ocultos sob um tema mais amplo.

2.2.3 Vetores e modelos

Na inferência das estruturas latentes de um corpus, é necessário um meio de representar documentos de forma numérica, mais prática para um computador. Uma abordagem comum é representar cada documento como um vetor de *features* (REHOREK, 2022). O autor explica que, cada *feature* pode ser um par de pergunta-resposta, como: quantas vezes a palavra “esponja” aparece no documento? Ou quantos parágrafos o documento possui? A representação deste documento se dá por uma série de pares, conhecidos como vetor denso. Se as perguntas são previamente conhecidas, elas podem ser deixadas implícitas, e o documento pode ser representado por uma sequência de respostas, resultando no vetor para o documento.

Supondo que as perguntas sejam as mesmas, pode-se comparar os vetores de dois documentos diferentes. A semelhança entre os vetores pode levar à conclusão de que seus documentos correspondentes também são semelhantes (MOSER *et al.*, 2009). Porém, a correção dessa conclusão depende da qualidade das perguntas iniciais.

Outra abordagem para representar um documento é o modelo *bag-of-words*. Cada documento é representado por um vetor contendo a contagem de frequência de

cada palavra no dicionário (ZHANG, Y.; JIN; ZHOU, Z.-H., 2010). O comprimento do vetor é o número de entradas no dicionário. O modelo *bag-of-words*, contudo, ignora completamente a ordem dos tokens no documento que está sendo codificado. Contudo, é importante ressaltar que dependendo de como a representação foi obtida, dois documentos diferentes podem ter as mesmas representações vetoriais (ZHANG, Y.; JIN; ZHOU, Z.-H., 2010). Uma vez que a ordem das palavras não é representada, os documentos "O gato perseguiu o rato" e "O rato perseguiu o gato" teriam a mesma representação vetorial no *bag-of-words*, por conterem as mesmas palavras com a mesma frequência.

Após a vetorização do corpus, inicia-se o processo de transformá-lo usando modelos (REHOREK, 2022). Compreende-se um modelo como uma transformação de uma representação de documento para outro. Nesse caso, como os documentos são representados por vetores, um modelo pode ser considerado uma transformação entre dois espaços vetoriais. Os detalhes dessa transformação são aprendidos pelo modelo durante o treinamento, quando ele lê o corpus de treinamento.

Um exemplo simples de modelo é o *Term Frequency–Inverse Document Frequency* (TF-IDF) (LI, Y.; SHEN, 2017). O modelo TF-IDF transforma vetores do modelo *bag-of-words* para um espaço vetorial onde as contagens de frequência são ponderadas conforme a raridade relativa de cada palavra no corpus. O modelo TF-IDF retorna uma lista de tuplas, onde a primeira entrada é o ID do token e a segunda entrada é o peso TF-IDF.

Uma vez que o modelo foi criado, é possível realizar uma série de operações com ele. Por exemplo, transformar todo o corpus via TF-IDF e indexá-lo, em preparação para consultas de similaridade (REHOREK, 2022), e consultar a similaridade de um documento específico em relação a todos os documentos no corpus.

2.2.4 Os primeiros modelos

A necessidade de descrever os elementos de uma grande coleção de dados impulsionou o desenvolvimento desse processo, preservando as relações estatísticas necessárias para concluir análises mais diretas, como classificação e síntese (BLEI; NG; JORDAN, 2003).

O primeiro método de modelagem de tópicos foi apresentado em 1980 como uma ramificação dos modelos generativos, abrangendo o campo probabilístico (LIU, L. *et al.*, 2016). Essa modelagem consiste em uma relação probabilística que gera dados em um conjunto, com base em como variáveis observáveis interagem com parâmetros latentes (STEYVERS; GRIFFITHS, 2007).

O primeiro método desenvolvido para realizar a modelagem de tópicos foi chamado de esquema TF-IDF (SALTON, 1983). Nesse método, considera-se o vocabulário de cada documento no corpus, e o número de ocorrências de cada palavra é armaze-

nado em um contador, para formar a frequência de termos TF específica para cada palavra de um documento. Calcula-se também a frequência inversa de documento IDF, ou seja, o total de instâncias de uma palavra em todo o corpus. Então, os valores TF e IDF são normalizados conforme o *data set*, e comparados para formar uma matriz termo-por-documento contendo os valores TF-IDF para todos os documentos (SALTON, 1983). Isso reduz o corpus em uma matriz dimensional *VxD* e os documentos em vetores de comprimento fixo, compostos por números reais e positivos. Este método mostrou-se eficaz para identificar conjuntos de palavras que distinguem documentos em uma coleção, porém a redução do comprimento de descrição não foi suficiente para produzir informações relevantes sobre as relações estatísticas dos documentos, seja dentro ou entre eles (BLEI; NG; JORDAN, 2003). Os autores ainda ressaltam que uma análise mais direta poderia ser implementada com um modelo generativo ajustado com métodos probabilísticos.

(HOFMANN, 1999) apresenta uma contribuição nesse caminho, chamada *Probabilistic Latent Semantic Analysis* (PLSA). Essa abordagem se afastou dos métodos de redução de dimensionalidade e se concentrou mais na modelagem probabilística. Todavia, essa abordagem possui um sério problema de *overfitting* conforme o volume do corpus aumenta. A falta de um modelo probabilístico para determinar as proporções de tópicos em um documento causa o aumento linear dos parâmetros totais do modelo em relação ao corpus. Em vista disso, o PLSA é incapaz de considerar documentos além do conjunto de treinamento (BLEI; NG; JORDAN, 2003).

Apesar das deficiências, essa foi a primeira instância de um método probabilístico sendo adotado para identificar tópicos. Eventualmente, essa abordagem levou às técnicas consideradas mais populares na atualidade, centradas em teoremas bayesianos, distribuições logísticas e fatoração de matrizes.

2.3 APRENDIZADO NÃO SUPERVISIONADO

O aprendizado não supervisionado é um campo do aprendizado de máquina que abrange, principalmente, algoritmos para analisar e clusterizar *data sets* não rotulados/indexados (HIRAN *et al.*, 2021). Os autores apontam que esses algoritmos descobrem padrões ocultos ou grupos de dados sem intervenção humana. A capacidade de identificar similaridades e diferenças na informação torna essa a solução ideal para análise de dados exploratória, segmentação de conteúdo, e reconhecimento de imagem.

Aplicações de aprendizado não supervisionado tornaram-se um método comum para aprimorar a experiência de usuários com produtos e assegurar a qualidade de sistemas (SUN *et al.*, 2019). A natureza do aprendizado fornece uma via exploratória para visualizar os dados, permitindo que padrões sejam identificados em grandes volumes de dados mais rapidamente que uma análise manual (HIRAN *et al.*, 2021).

Dentre aplicações de aprendizado não supervisionado, destacam-se:

- **selecionar documentos**: categorização de dados textuais conforme a similaridade do conteúdo ou características dos documentos (BISANDU; PRASAD; LIMAN, 2018);
- **visão computacional**: tarefas de percepção visual, como reconhecimento de objetos (CARON *et al.*, 2018);
- **imagens médicas**: análise de imagens médicas, agindo na detecção, classificação e segmentação de imagens (LATIF *et al.*, 2019). Na radiologia e patologia, pode auxiliar o diagnóstico de pacientes com rapidez e precisão (LEE, J.-G. *et al.*, 2017);
- **detecção de anomalia**: identificar pontos de dados atípicos em um conjunto de dados (NASSIF *et al.*, 2021). Detectar anomalias contribui na conscientização sobre equipamentos defeituosos, erro humano ou violações de segurança;
- **segmentação de clientes**: definir personas de clientes para compreender os traços comuns e hábitos de compra destes (NILASHI *et al.*, 2021). Assim, criam-se perfis de persona de compradores, permitindo que as organizações alinhem seus produtos segundo as necessidades de seus compradores;
- **sistemas de recomendação**: descobrir tendências de dados que podem ser usadas para desenvolver estratégias de recomendação mais eficazes (SZABÓ; GENGE, 2020). Baseado em dados de avaliações, varejistas online adotam essa solução para prever ações favoráveis de usuários, como uma compra.

Apesar dos muitos usos e benefícios do aprendizado não supervisionado, existem barreiras para executar modelos de aprendizado de máquina sem intervenção humana. Quando se trabalha com grandes volumes de dados, a complexidade computacional para tarefas essas tarefas de aprendizado é notável e, consequentemente, o tempo de treinamento do modelo aumenta (SCHMARJE *et al.*, 2021). Além disso, uma máquina totalmente independente tem alto risco de apresentar resultados imprecisos (CHANDRA; HAREENDRAN *et al.*, 2021). Em razão disso, é preciso uma intervenção humana para validar as variáveis de saída, mesmo com a falta de transparência sobre como o algoritmo agrupa os dados.

Considerando as características dos modelos de aprendizado não supervisionado, existem três tarefas em que eles são frequentemente empregados: clusterização, associação, e redução de dimensionalidade.

2.3.1 Associação

Uma regra de associação é um método baseado em regras para encontrar relações entre variáveis de um *data set* (GHAFARI; TJORTJIS, 2019). Métodos dessa natureza são comumente usados por organizações para entender melhor as relações entre produtos distintos. Com isso, os hábitos de consumo dos clientes podem ser compreendidos, visando desenvolver recomendações e ofertas mais precisas (WU, P.-J.; LIN, 2018).

Os algoritmos apriori (AGRAWAL; IMIELIŃSKI; SWAMI, 1993) são um exemplo de regra de associação. Estes foram popularizados por meio de análises de mercado que geravam mecanismos de recomendação para plataformas de música e e-commerce (GHAFARI; TJORTJIS, 2019). Eles são utilizados em bases de dados transacionais para identificar interações frequentes entre itens. Dessa forma, identifica-se a probabilidade de consumo de um produto, dado o consumo de outro produto (DAS *et al.*, 2021). Nesse caso, um usuário que realiza uma compra online de um laptop, receberá recomendações de acessórios para laptop, como capas case, bolsas para laptops ou mouses. Isso é baseado em hábitos de compra anteriores, bem como nos de outras pessoas.

2.3.2 Redução de dimensionalidade

Em aprendizado não supervisionado, normalmente, mais entradas produzem resultados mais precisos. Contudo, isso impacta o desempenho dos algoritmos de aprendizado, causando problemas como *overfitting*, que dificultam a visualização do *data set* (PATEL, 2019). Em geral, a redução de dimensionalidade é uma técnica usada na etapa de pré-processamento, quando o número de características, ou dimensões, em um conjunto de dados é muito alto (HUANG, X.; WU, L.; YE, 2019). Com ela, a quantidade de inputs de dados é reduzida para um número gerenciável enquanto preserva o máximo possível da integridade do *data set* (PATEL, 2019).

A Análise de Componentes Principais (PCA) é um algoritmo de redução de dimensionalidade usado para diminuir redundâncias e para compactar *data sets* por meio de extração de características (KARAMIZADEH *et al.*, 2013). Esse método usa uma transformação linear para criar uma nova representação dos dados, gerando um conjunto de “componentes principais”.

Conforme Patel (2019), esse processo se repete segundo o número de dimensões, onde o primeiro componente é a direção que maximiza a variância do conjunto de dados. Já o segundo componente também busca a variância máxima nos dados, mas sem correlação ao primeiro componente principal. Isso acontece, pois, o segundo produz uma direção perpendicular ou ortogonal ao primeiro componente.

2.3.3 Clusterização

A clusterização é uma técnica de mineração de dados que agrupa elementos não rotulados baseados em suas similaridades e diferenças (XU, D.; TIAN, 2015). Os autores destacam que esses algoritmos são usados para processar unidades de dados, brutos e não classificados, em grupos que representam suas estruturas ou padrões de informação. Dentre as categorias de algoritmos de clusterização, quatro delas se destacam: exclusiva, sobreposta, hierárquica, e probabilística.

2.3.3.1 Clusterização exclusiva e sobreposta

Clusterização exclusiva é uma forma de agrupamento que considera que uma unidade de dado existe unicamente em um cluster (JIPKATE; GOHOKAR, 2012). O algoritmo *K-means* (HARTIGAN, 1975) é um exemplo de clusterização exclusiva. No *K-means*, os dados são dispersos em K grupos, baseados na distância do centroide de cada cluster (LIKAS; VLASSIS; VERBEEK, 2003). Os dados mais próximos de um determinado centroide serão agrupados sob a mesma categoria. Um valor maior de K é um indicativo de agrupamentos menores com mais granularidade; enquanto um valor menor de K representa grupos maiores e menos granularidade (LIKAS; VLASSIS; VERBEEK, 2003). *K-means* pode ser aplicado em contextos como segmentação de mercado (HUNG; NGOC; HANH, 2019), clusterização de documentos (WU, G. *et al.*, 2015), segmentação de imagens (ZHENG *et al.*, 2018), e compressão de imagens (WAN, 2019).

O diferencial dos clusters sobrepostos para os exclusivos é que estes admitem que os dados pertençam a múltiplos clusters com graus diferentes de filiação (JIPKATE; GOHOKAR, 2012). *Soft K-means* ou *Fuzzy K-means* é um exemplo de clusterização sobreposta (XU, D.; TIAN, 2015).

2.3.3.2 Clusterização hierárquica

A clusterização hierárquica, é um algoritmo não supervisionado de agrupamento que pode ser categorizado de duas formas: aglomerativos ou divisivos (MURTAGH; CONTRERAS, 2012). A clusterização aglomerativa é considerada uma abordagem "*bottoms-up*". A princípio, os dados são isolados em agrupamentos separados, e então são mesclados iterativamente de acordo com as semelhanças, até formar um cluster.

Por sua vez, a clusterização divisiva adota uma abordagem "*top-down*", definida como o oposto da aglomerativa. Nesse caso, os clusters de dados são divididos conforme as diferenças das unidades de dados. Essa abordagem não é regularmente usada, mas ainda é relevante no contexto da clusterização hierárquica.

2.3.3.3 Clusterização probabilística

Um modelo probabilístico pode ser empregado para estimar a densidade de um cluster (AHMAD; KHAN, 2019). O *Gaussian Mixture Model* (GMM) é um dos métodos probabilísticos de cluster mais usados.

Os modelos *Gaussian Mixture* são feitos de um número não especificado de funções de distribuição de probabilidade (MCLACHLAN; RATHNAYAKE, 2014). Os autores ressaltam que os GMMs são usados principalmente para determinar a qual distribuição de probabilidade, gaussiana ou normal, um dado pertence. Se a média ou variância são conhecidas, então determina-se a qual distribuição a unidade de dado pertence. Todavia, nos GMMs essas variáveis são desconhecidas, portanto, assume-se que uma variável oculta existe para agrupar os dados corretamente (AHMAD; KHAN, 2019).

2.4 MÍDIAS SOCIAIS

Mídia social é um termo coletivo para sites e aplicativos voltados para comunicação baseada em comunidades, colaboração, interação, e compartilhamento de conteúdo (HJORTH; HINTON, 2019). Em menos de uma geração, as mídias sociais evoluíram de um espaço exclusivo para troca direta de informações eletrônicas, para um local de encontro virtual (TUTEN, 2020).

As mídias sociais têm grande alcance global. Os aplicativos em dispositivos móveis tornam essas plataformas altamente acessíveis (YADAV; JOSHI; RAHMAN, 2015). Alguns exemplos populares de mídia social incluem Twitter, Facebook e LinkedIn. Os autores descrevem as diferentes categorias de plataformas sociais, dentre elas quatro se destacam:

- **Redes sociais.** As pessoas usam essas plataformas para se conectarem e compartilharem informações, pensamentos e ideias. O foco dessas redes geralmente está no usuário. Os perfis pessoais ajudam os participantes a identificar outros usuários com interesses ou preocupações comuns. Facebook e LinkedIn são exemplos de redes sociais.
- **Redes de compartilhamento de mídia.** O foco dessas redes está no conteúdo. No YouTube, a interação está em torno dos vídeos que os usuários criam. Outras redes de compartilhamento de mídia são TikTok e Instagram. Plataformas de streaming como a Twitch são consideradas um subconjunto desta categoria.
- **Redes baseadas em comunidade.** O foco desse tipo de rede social é a discussão aprofundada, bem como um fórum de blog. Os usuários discutem assuntos diversos em publicações com comentários detalhados. As comuni-

dades geralmente se formam em torno de assuntos selecionados. O Reddit é um exemplo de rede baseada em comunidade.

- **Redes de comitês de revisão.** Nessas redes, o foco é uma revisão, geralmente de um produto ou serviço.No Yelp, por exemplo, os usuários podem escrever avaliações sobre restaurantes e endossar as opiniões uns dos outros para aumentar a visibilidade.

A sociedade moderna utiliza cada dia mais a internet para uma ampla variedade de tarefas, que incluem reunir, compartilhar e comentar sobre conteúdos, eventos e acontecimentos (PAPASAVVA *et al.*, 2020). Os usuários das redes online podem publicar conteúdos sem passar por qualquer processo editorial, revisão por pares, verificação de qualidade ou citação de uma fonte qualquer (HOSSAIN *et al.*, 2020). Os autores ainda alertam que essas características tornam essas redes um ambiente propício para a disseminação de desinformação, que se soma à tendência natural das mídias sociais para polarização. Assim, formam-se bolhas sociais com comunidades de usuários isolados de ideias e opiniões contrárias às deles. É comum que ambientes polarizados promovam a disseminação de desinformação com a intenção de enganar outros com informações falsas.

O engajamento de uma mídia social consiste nas várias maneiras pelas quais os usuários interagem com uma publicação (HJORTH; HINTON, 2019). Isso pode incluir comentários, seguidores, compartilhamentos (retuítes no Twitter) e cliques em um link compartilhado. Além disso, muitas empresas usam dessas plataformas sociais para comercializar e promover seus produtos enquanto acompanham as preocupações dos clientes (TUTEN, 2020). Também é comum a prática do crowdsourcing, isto é, usar as redes sociais para reunir conhecimento, bens ou serviços (GAO, H.; BARBIER; GOOLSBY, 2011; SIMULA; TÖLLINEN; KARJALUOTO, 2013).

Enquanto usuários interagem por espaços de mídias sociais, eles formam conexões que se manifestam em estruturas sociais complexas. Essas conexões que indicam como um conteúdo está sendo compartilhado e como isso reflete no fluxo das informações sendo compartilhadas (HIMELBOIM *et al.*, 2017). Nesse contexto, a mineração de dados de mídias sociais e processamento de linguagem natural, por exemplo, podem revelar informações relevantes para fins como: auxiliar a construção de uma campanha de comunicação sobre saúde pública (SIDDIQUI; RATHINAM, 2021), análise de sentimento durante um período de eleição presidencial (ALASHRI *et al.*, 2016), ou monitorar o uso de termos emergentes para drogas (SIMPSON *et al.*, 2018).

A breve história das mídias sociais mostra como a popularização do acesso, e a influência na dinâmica cultural transformaram rapidamente o cenário e moldaram as tecnologias.

2.5 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as definições teóricas que embasam o desenvolvimento deste trabalho. A compreensão das características básicas de um algoritmo de modelagem de tópicos, como aprendizado não supervisionado e clusterização, são fundamentais para compreensão da técnica de modelagem de tópicos.

O domínio de aplicação da modelagem de tópicos pode abranger diversos tipos de dados como textos, vídeos ou imagens. Este trabalho compreende a aplicação da modelagem de tópicos em textos provenientes de uma plataforma de mídia social. Para tanto, foram descritos aspectos pertinentes de tais plataformas, deixando em evidência o valor que esses dados possuem.

3 TÉCNICAS DE MODELAGEM DE TÓPICOS

A modelagem de tópicos é uma técnica que junta PLN e aprendizado não supervisionado, e já demonstrou ser efetiva para analisar grandes quantidades de texto. Uma vasta literatura sobre modelagem de tópicos existe, com modelos propostos desde a década de 1990 (DEERWESTER *et al.*, 1990; HOFMANN, 1999; LEE, D. D.; SEUNG, 1999). Entretanto, como já discutido, nem todos os modelos são adequados para o uso com textos curtos, principalmente devido a esparsidade que esses dados apresentam.

Este capítulo apresenta as quatro categorias utilizadas para classificar as técnicas de modelagem. Da mesma forma, apresentam-se as características observadas em cada método reunido para esse trabalho, uma discussão sobre as propriedades destes métodos, e um quadro comparativo que lista as principais características dos algoritmos, mostrando as configurações e parâmetros a serem adotadas para este estudo.

3.1 MODELOS TRADICIONAIS

Os algoritmos tradicionais de modelagem de tópicos como LDA conseguem identificar, em textos, os conjuntos de palavras que possuem alguma relação entre si e classificá-las em conjuntos de acordo com essa similaridade (QIANG *et al.*, 2020), onde cada conjunto é considerado um tópico. Esses foram as primeiras soluções propostas para modelagem de tópicos, em uma época anterior a popularização das mídias sociais.

No entanto, quando aplicados em mídias sociais, os modelos Tradicionais apresentam alguns desafios. Isso ocorre porque os textos publicados nas mídias sociais são tipicamente curtos, com um contexto limitado. Isso dificulta a identificação de padrões e temas significativos por parte das técnicas, que não possuem nenhuma solução visando o tamanho reduzido dos textos. Dessa forma, a esparsidade dos dados textuais fornecem pouca informação sobre o tema dos documentos, e, por consequência, os textos nas mídias sociais são tipicamente desconexos e escassos de informação sobre a situação em que foram publicados.

A Tabela 1 mostra um exemplo de aplicação da técnica LDA de modelagem de tópicos em textos curtos de mídias sociais. Um corpus formado por 14.739 tuítes brasileiros relacionados à COVID-19, do dia 21/06/2021. Antes da execução do LDA, os textos passaram por três etapas de pré-processamento: (1) a tokenização, onde todos os tuítes tiveram suas palavras segmentadas, ou seja, a sequência de palavras foi quebrada com base nos limites de cada palavra; (2) o *stopping*, onde as *stop words* foram removidas dos documentos; e (3) stemização, cada palavra foi reduzida até seu radical (por exemplo, “vacina” e “vacinou” foram reduzidas à “vacin”).

Tabela 1 – Tópicos identificados a partir dos 14.739 tuítes em português.

Tópicos	**Palavras**									
Tópico 1	covid	casos	morte	mais	vacina	brasil	para	pelo	contra	quem
Tópico 2	covid	vacina	para	contra	mais	brasil	dose	pelo	todo	pessoa

Para este exemplo, determinou-se que dois tópicos deveriam ser detectados pelo algoritmo. Fica evidente que alguns termos aparecem repetidos entre os dois tópicos, além de palavras sem nenhuma relação aparente com as outras. Isso se dá por duas limitações: a primeira é pela baixa quantidade de documentos no corpus, pois, mesmo 14.739 tuítes geram um volume de texto pequeno para qualquer técnica de modelagem. O segundo motivo é a técnica utilizada. Nesse caso, o resultado do LDA é afetado pelo tamanho dos documentos usados para construir o modelo (WU, X.; LI, Chunping, 2019). Os textos curtos, comumente gerados nas mídias sociais, são esparsos e limitados em contexto, com menos ocorrências de pares de palavras com similaridade semântica (WU, X. *et al.*, 2020).

Dentre os modelos tradicionais escolhidos para esta pesquisa, estão o *Latent Dirichlet Allocation* (LDA), *Latent Semantic Analysis* (LSA), *Probabilistic Latent Semantic Analysis* (PLSA), e *Non-negative Matrix Factorization* (NMF).

3.1.1 *Latent Dirichlet Allocation* (LDA)

Um dos métodos pioneiros de modelagem de tópicos, e frequentemente empregado, é o Latent Dirichlet Allocation, ou LDA, introduzido por Blei, Ng e Jordan (2003). Essa abordagem divide os dados em três níveis: palavra, tópico e documento. O LDA trata os documentos como uma grande mistura de diversos tópicos que, por sua vez, são uma distribuição de probabilidades sobre palavras.

O LDA inicia com a atribuição de dois parâmetros Dirichlet: um para a distribuição de tópicos por documento e outro para a distribuição de palavras por tópico. Por isso cada documento na coleção é representado como uma mistura de tópicos, e cada palavra é atribuída a um tópico. O LDA assume que as palavras de cada documento são intercambiáveis (ou seja, a ordem das palavras não importa), seguindo o conceito de *bag-of-words*. O processo é iterativo e, a cada iteração, o modelo revisa cada palavra em cada documento, ajustando a atribuição de tópicos com base na probabilidade de a palavra pertencer a um tópico, dado o tópico dos documentos.

Para gerar cada documento, primeiro tira-se uma amostra de um K-vetor apresentando a proporção de mistura de tópicos de uma Distribuição de Dirichlet $p(\theta|\alpha)$. A variável k definirá a dimensão dessa distribuição, e consequentemente, também a dimensão da variável de tópico z, mas também representa o número total de tópicos que serão retornados no modelo.

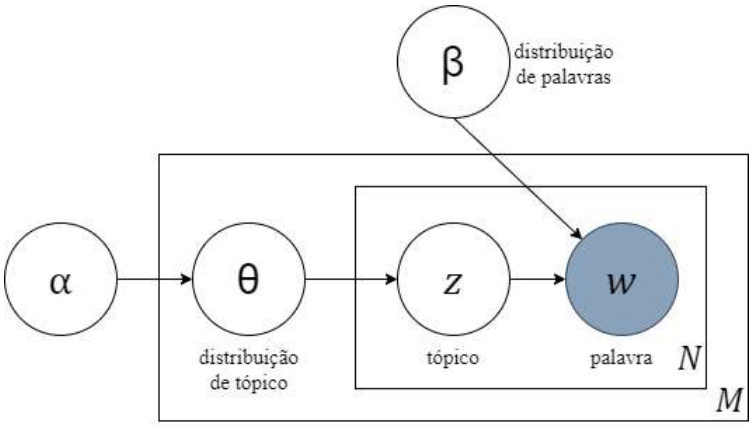
Adicionalmente, a matriz β com dimensões $k \times V$ parametriza as probabilidades de palavras tal que $\beta_{ij} = p(w^j = 1|z^i = 1)$ onde $i = 0,1,...,K$ e $j = 0,1,...,V$. Onde $\theta_i \geq 0$ $\sum_{i=1}^{k} \theta_i = 1$, uma variável Dirichlet θ de dimensionalidade k pode ocupar valores no simplex (k-1) e tem uma densidade de probabilidade neste simplex determinada pela seguinte equação:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{z_1-1} ... \theta_k^{z_k-1} \tag{1}$$

O parâmetro na Equação (1) é o hiperparâmetro da distribuição Dirichlet. Este valor é como uma contagem das vezes que um tópico individual foi observado em um documento, incorporado por um k-vetor de elementos $\alpha_i > 0$. As combinações de tópicos são influenciadas pelo valor desse parâmetro. Para conteúdo de mídias sociais, recomenda-se manter $\alpha < 1$, pois isso localiza os modos da distribuição de Dirichlet nos cantos do simplex e cria uma tendência para a esparsidade (STEYVERS; GRIFFITHS, 2007). Os autores ainda ressaltam que o hiperparâmetro pode ser interpretado como o número de ocorrências de palavras amostradas antes da observação de qualquer palavra dentro do corpus.

O processo generativo de um modelo de mistura típico, como o do LDA, envolve a geração de documentos a partir de uma mistura de tópicos, onde cada tópico é uma distribuição sobre um vocabulário fixo. Portanto, a Figura 2 ilustra o processo generativo básico para este modelo.

Figura 2 – Representação gráfica do modelo LDA.



Fonte: Adaptado de Blei, Ng e Jordan (2003).

O modelo LDA foi ilustrado na Figura 2 no formato de “notação de placa”. As caixas da Figura são chamadas de placas e indicam ações replicadas no modelo. Os círculos representam parâmetros, sendo que os brancos indicam que a variável é latente ou oculto, enquanto o escuro indica que informação foi fornecida. As setas

indicam a influência de uma variável sobre a outra. Nesse caso, cada amostra de palavra é selecionada de forma independente e individual da distribuição para cada tópico.

Essa abordagem do LDA mostrou uma melhora significativa em relação a outros trabalhos, por considerar a inferência de documentos novos e o entendimento de dados não estruturados. Entretanto, várias etapas de testes são necessárias para estimar bons valores para e a fim de maximizar a probabilidade marginal de registro dos dados. Além disso, o LDA mostrou problemas com a esparsidade dos dados quando há um amplo vocabulário no corpus (BLEI; NG; JORDAN, 2003; VORONTSOV; POTAPENKO, 2014).

Assim sendo, a abordagem do LDA apresenta adversidades como inferência de parâmetros e instabilidades nos resultados. Dentre elas está a quantidade de tópicos K. Essa variável representa a estrutura de tópicos a ser gerada pelo modelo, portanto, tem impacto significativo em como os tópicos são formados (ALSUMAIT *et al.*, 2009). Dificilmente há como estimar a quantidade ideal de tópicos antes gerar um modelo. Por isso é comum que pesquisadores gerem tópicos iterativamente com os mesmos dados, enquanto usam valores diferentes para K e então analisem os resultados usando certas métricas como a perplexidade. Um problema dessa solução é que esse processo exige mais tempo e poder computacional conforme cresce o volume do *data set*.

Outro problema para essa abordagem é a grande correlação entre o conteúdo dos diversos documentos, principalmente quando se trata de textos extraídos de mídias sociais. Isso limita a capacidade do algoritmo de lidar com o *big data* e fazer predições corretas sobre novos documentos, que, por sua vez, reduz as aplicações em cenários reais (CHENG *et al.*, 2014). A ordem das palavras no documento, e as relações dos documentos ao longo do tempo são características importantes de dados textuais e linguísticos advindos de cenários reais (BLEI; LAFFERTY, 2006). Por se tratar de um obstáculo inerente da distribuição Dirichlet, não há quantidade adicional de passos ou otimizações que possam contornar essa falha do LDA (BLEI; NG; JORDAN, 2003).

Assim como outros métodos, o LDA é propenso a variações nos resultados, proporcional a ordem dos dados utilizados para treinamento (HADI; FARD, 2020). Com isso, o modelo cria tópicos diferentes em cada iteração com os mesmos dados, e atribui palavras diferentes para tópico semelhantes em cada modelo. Nesse caso, utiliza-se uma prática comum na análise de dados: usar apenas uma parte do data set para realizar o treinamento do modelo. Ainda assim, fica claro como os testes iterativos são inefetivos para otimizar tópicos, e os modelos LDA são inconstantes.

O método tradicional do LDA e suas variações foram utilizadas para diversos propósitos, fazendo deste o método mais popular em pesquisas sobre modelagem de tópicos (BLEI, 2012). Ele é propenso a falhas e requer otimizações para alcançar bons resultados; porém, é adaptável em aplicações variadas com diferentes tipos de

dados e versátil com variações do algoritmo que buscam suprir as dificuldades do original. Além disso, a grande popularidade do LDA o torna um elemento importante da modelagem de tópicos, e por isso é comumente usado para fins de aferição de outros modelos, servindo como linha de base para medir o desempenho de outras técnicas.

3.1.2 *Latent Semantic Analysis* (LSA)

O LSA, é a uma das técnicas pioneiras de modelagem de tópicos (DEERWESTER *et al.*, 1990). Segundo o autor, a ideia principal é construir uma matriz com o que se tem (documentos e termos) e decompô-la em duas matrizes distintas: documento-tópicos e tópico-termos.

O LSA, também conhecida como *Latent Semantic Indexing*, é uma técnica de mineração de texto e PLN feita para identificar relações semânticas latentes entre termos e documentos em um corpus. Esse algoritmo tem sido amplamente empregado na detecção de tópicos. O principal objetivo do LSA é reduzir a dimensionalidade dos dados textuais, enquanto mantém a estrutura semântica subjacente.

O primeiro passo é gerar a matriz documento-termo. Dado os m documentos e n termos no vocabulário, pode-se construir uma matriz $A_{m \times n}$ onde cada linha representa um documento e cada coluna representa uma palavra. Na versão mais básica do LSA, cada registro na matriz pode apenas representar o número de vezes que a j-ésima palavra aparece no i-ésimo documento. Contudo, na prática, apenas uma contagem não é suficiente para considerar o significado de cada palavra no documento. Por exemplo, a palavra “vacina” informa mais sobre o tópico presente em um documento que a palavra “teste”.

Por esse motivo, no lugar de contadores de frequência, os modelos LSA adotaram nas matrizes termo-documento uma pontuação conhecida como TF-IDF. Essa é uma medida estatística que indica a importância da palavra em um documento em relação a uma coleção de documentos (ANAND; JEFFREY DAVID, 2011). Nesse caso, o TF-IDF designa um peso para o termo j no documento i, como mostra a Equação (2):

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_j} \tag{2}$$

Na Equação (2), w representa a pontuação TF-IDF da palavra j e documento i; tf reflete o número de ocorrências de um termo j em um documento i; N é o total de documentos; e df representa a quantidade de documentos que contem o termo j. Portanto, a Equação (2) mostra que j possui maior pontuação TF-IDF quando aparece com mais frequência em i e com menos frequência ao longo do corpus (ANAND; JEFFREY DAVID, 2011).

Uma palavra como “coronavírus” que aparece com frequência num documento,

e também é comum no resto do corpus, vai ter baixa pontuação TF-IDF. Por outro lado, uma palavra frequente num documento, mas rara no resto do corpus, possuirá pontuação TF-IDF mais alta.

Uma vez construída a matriz documento-termo *A*, pode-se considerar os tópicos latentes presentes nela. Entretanto, *A* é muito esparsa, possui muito ruídos, e muitas redundâncias em suas dimensões (DEERWESTER *et al.*, 1990). À vista disso, os autores para encontrar os poucos tópicos latentes que capturam as relações entre termos e documentos, realiza-se uma redução de dimensionalidade em *A*.

Tal redução de dimensionalidade pode ser feita com o *Singular Value Decomposition* (SVD) truncado. O SVD é uma técnica de álgebra linear que fatora qualquer matriz *M* nos produtos de 3 matrizes separadas: $M = U \times S \times V$, onde *S* é a matriz diagonal dos valores singulares de *M* (HANSEN, 1987). Sendo assim, o SVD truncado reduz a dimensionalidade ao selecionar apenas os *t* maiores valores singulares. Mantém-se apenas as *t* primeiras colunas de *U* e *V*. Nesse caso, o autor esclarece que *t* é um hiperparâmetro ajustável, e reflete o número de tópicos que busca-se formar, conforme mostra a equação (3):

$$A = U_t S_t V_t^T \tag{3}$$

Dessa forma, são mantidas no espaço transformado apenas as *t* dimensões mais significativas. Assim, *U* emerge como a matriz documento-tópico que mostra a associação entre cada documento e os tópicos; *V* torna-se a matriz termo-tópico que mostra a força da associação entre cada palavra e os tópicos; e *S* representa a matriz diagonal que avalia a "força" de cada tópico no corpus. Em *U*, *V* e *S* as colunas correspondem a um dos *t* tópicos.

A matriz *U* possui dimensões $m \times t$, onde as linhas representam os vetores de documentos m, com dimensões; a matriz *V* possui dimensões $n \times t$, no qual as linhas refletem os vetores de termos *n*; enquanto *S* possui dimensões $t \times t$.

Com os vetores de documentos e de termos, aplica-se medidas como a similaridade de cossenos para avaliar: a similaridade de diferentes documentos; similaridade de palavras diferentes; e a similaridade entre termos e documentos (DEERWESTER *et al.*, 1990).

Uma limitação do LSA é que esse considera apenas a ocorrência linear de palavras em documentos, o que pode não ser suficiente para capturar todas as nuances semânticas e relacionamentos entre termos. Além disso, o LSA é um método de aprendizado não supervisionado, o que significa que os tópicos gerados podem ser difíceis de interpretar e rotular.

Ademais, apesar de ser rápido e eficiente para o uso, o LSA possui algumas desvantagens como: a falta de eficiência para representações, e a necessidade de um grande conjunto de documentos e vocabulário para resultados mais precisos (HOF-

MANN, 1999). Além disso, a implementação da medida TF-IDF prejudica os resultados quando o corpus é formado por textos curtos.

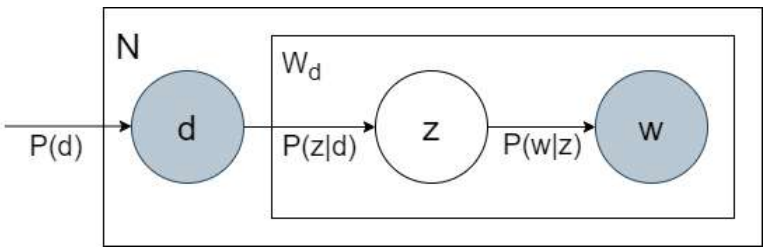
3.1.3 *Probabilistic Latent Semantic Analysis* (PLSA)

Enquanto o LSA utiliza o SVD para resolver o problema de redução de dimensionalidade, o PLSA usa um método probabilístico para resolver o problema de redução de dimensionalidade (HOFMANN, 1999). O método visa encontrar um modelo probabilístico com tópicos latentes que geram os dados observáveis na matriz documento-termo.

A ideia central do PLSA é a suposição de que as palavras nos documentos são geradas por uma mistura finita de tópicos, onde cada tópico é uma distribuição multinomial de palavras. Assim como o LDA, cada documento é representado como uma mistura de tópicos. No entanto, diferentemente do LDA, que assume uma distribuição Dirichlet para a mistura de tópicos em documentos e para a distribuição de palavras nos tópicos, o PLSA modela diretamente a probabilidade condicional de uma palavra dada a presença de um tópico e de um tópico dado um documento.

Assim, tem-se um modelo $P(D, W)$, sendo que para cada documento d ou termo w, $P(d, w)$ corresponda a um registro na matriz documento-termo (HOFMANN, 1999). Nesse caso, dado que cada documento consiste de uma mistura de tópicos, e cada tópico consiste em uma coleção de palavras, o PLSA explora essa afirmativa de um ponto de vista probabilístico, conforme ilustra a Figura 3.

Figura 3 – Representação gráfica do modelo LDA.



Fonte: Adaptado de Blei, Ng e Jordan (2003).

Conforme mostrado na Figura 3, dado um documento d, um tópico z está presente nesse documento com probabilidade P($z|d$). Ademais, dado um tópico z, um termo w pode ser extraído de z com probabilidade $P(w|z)$. Portanto, formalmente a probabilidade conjunta de um determinado documento e termo estarem juntos pode ser representada como:

$$P(D,W) = P(D) \sum_{Z} P(Z|D)P(W|Z) \tag{4}$$

O segundo membro (lado direito) da equação mostra a probabilidade de um documento, e então, baseado na distribuição de tópicos daquele documento, a probabilidade de encontrar uma certa palavra dentro dele. Dessa forma, $P(D)$, $P(Z|D)$ e $P(W|Z)$ são parâmetros para o modelo.

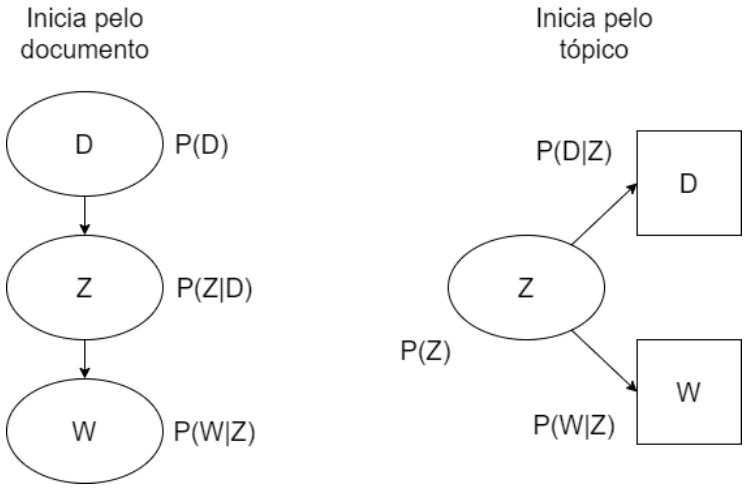
$P(D)$ pode ser determinado diretamente do corpus. Já $P(Z|D)$ e $P(W|Z)$ são modelados como distribuições multinomiais, treinados com o algoritmo *Expectation-Maximization* (EM). O EM é um algoritmo usado para encontrar as estimativas de parâmetros prováveis para um modelo que depende de variáveis latentes não observadas (MOON, 1996), que nesse caso são os tópicos.

Alternativamente, $P(D, W)$ também pode ser representado usando um conjunto diferente de três parâmetros:

$$P(D,W) = \sum_{Z} P(Z)P(D|Z)P(W|Z) \tag{5}$$

Isso se dá, pois o modelo funciona como um processo generativo (HOFMANN, 1999). Na parametrização mostrada na equação (4), inicia-se com um documento $P(D)$, gera-se o tópico com $P(Z|D)$, e o termo com $P(W|Z)$. Na parametrização da equação (5), inicia-se com o tópico $P(Z)$, e então gerando $P(D|Z)$ independentemente do termo $P(W|Z)$. A Figura 4 representa visualmente os dois modelos.

Figura 4 – Representação das parametrizações do PLSA.



Fonte: Adaptado de Blei, Ng e Jordan (2003).

Com essa alternativa de parametrização, é possível observar um paralelo direto do modelo PLSA com o modelo LSA. A probabilidade do tópico $P(Z)$ corresponde à matriz diagonal de probabilidades de tópicos singulares, $P(D|Z)$ corresponde à matriz documento-tópico U, e $P(W|Z)$ corresponde à matriz termo-tópico V.

No fim, apesar de distinções e de abordar o problema de uma forma diferente, o PLSA segue a fórmula do LSA, porém adiciona um tratamento probabilístico de tópicos e palavras na solução original (HOFMANN, 1999). Ele mostra-se como um modelo bem mais flexível, porém ainda apresenta alguns problemas, especialmente quanto ao número de parâmetros (BLEI; NG; JORDAN, 2003).

O número de parâmetros para PLSA cresce linearmente com o número de documentos; por isso, é propenso a *overfitting*. Além disso, num corpus de grande volume torna-se computacionalmente intratável devido as enormes matrizes que se criam para o modelo. Do mesmo modo, isso leva a estimativas instáveis que causam máxima local.

O LDA foi proposto como uma extensão do PLSA, mantendo as vantagens enquanto resolve os diversos problemas do método.

3.1.4 *Non-negative Matrix Factorization* (NMF)

Apresentado por (LEE, D. D.; SEUNG, 1999), o NMF mostra-se como uma alternativa para o LDA, apesar de precedê-lo. Ainda assim, esse método mostrou que consegue trabalhar efetivamente extraindo tópicos de um corpus de texto.

Diferente do LDA, por exemplo, que pode ser supervisionado ou não, o NMF é inteiramente não supervisionado. Esse utiliza métodos de redução de dimensionalidade para matrizes não-negativas (LEE, D. D.; SEUNG, 1999). Tal abordagem é similar a outras das quais se derivou a modelagem de tópicos, como a redução TF-IDF (SALTON, 1983) ou o LSA (DEERWESTER *et al.*, 1990).

Na modelagem de tópicos, o NMF é aplicado à matriz termo-documento, que é decomposta em duas matrizes menores: uma matriz de tópicos-palavras e uma matriz de documentos-tópicos. Cada tópico é então representado como uma combinação linear de palavras, e cada documento é representado como uma combinação linear de tópicos. Uma das principais vantagens do NMF é que ele fornece uma representação esparsa e parte por parte dos dados, o que pode facilitar a interpretação dos tópicos.

Nesse caso, considera-se no corpus os termos V e os documentos D para formar uma matriz termo-documento de dimensão $V \times D$. Com isso, aproxima-se esta matriz como o produto de dois fatores não negativos e k-dimensionais, contendo os pesos de seus componentes, que podem ser chamados de W e H (LEE, D. D.; SEUNG, 1999; BELFORD; MAC NAMEE; GREENE, 2018).

As linhas de H são como os pesos para cada respectivo termo das V palavras contidas no vocabulário do corpus, representando K tópicos, dando-lhe uma dimensionalidade de $k \times V$. As colunas de W representam então os pesos relativos a cada um dos documentos D em relação a cada tópico, dando-lhe uma dimensionalidade de $D \times k$. Ao colocar em ordem as linhas de H, forma-se uma classificação dos termos em relação ao tópico daquela linha, fornecendo assim um descritor de tópico.

O método é inicializado com uma atribuição aleatória de pesos aos elementos de *W* e *H*, e então estes fatores são melhorados iterativamente aplicando um algoritmo de otimização que reduz o erro de aproximação até que o processo atinja um mínimo localizado (BELFORD; MAC NAMEE; GREENE, 2018).

No entanto, assim como outras técnicas de modelagem de tópicos, o NMF tem suas limitações. O NMF não considera a estrutura sequencial dos dados, o que pode ser um problema ao lidar com textos onde a ordem das palavras é importante. Além disso, como o NMF é baseado em uma abordagem determinística, ele pode não ser tão robusto quanto os métodos probabilísticos, como o LDA e o PLSA, frente a dados ruidosos ou inconsistentes.

O autor ainda salienta que, embora o NMF tenha menos parâmetros configuráveis para o processo de modelagem e tenha demonstrado distinguir tópicos mais realistas do que o LDA, os valores iniciais para os pesos *H* e *W* têm grande influência nos resultados durante a otimização. Tal comportamento mostrou-se persistente pelas iterações do modelo. Como esses valores são atribuídos aleatoriamente, isso resulta em altos níveis de imprecisão e falta de reprodutibilidade para os fatores finais, especialmente para textos curtos.

3.2 MODELOS PARA TEXTOS CURTOS

Quando se trata de textos curtos, como os que são comumente encontrados em publicações de mídias sociais, técnicas tradicionais como o LDA sofrem com a esparsidade dos dados, isto é, a co-ocorrência de palavras torna-se menos comum devido ao curto tamanho dos textos (WU, X.; LI, Chunping, 2019; HADI; FARD, 2020). Os tópicos gerados a partir de textos curtos tendem a ficar repetitivos e incluir muitos ruídos, que atrapalham o resultado final da modelagem (WU, X. *et al.*, 2020).

Na literatura existem alternativas para este e outros problemas/limitações, com algoritmos e procedimentos de pré-processamento dos dados que mostram resultados mais eficazes ao se tratar de modelagem de tópicos em textos curtos. Qiang *et al.* (2020) consideram que as técnicas para modelagem de tópicos se dividem em três categorias: modelos baseados em DMM, auto-agregação, e co-ocorrência global de palavras. A seguir serão apresentadas estas categorias e alguns de seus modelos que se destacaram na bibliografia.

3.2.1 Baseados em DMM

O modelo DMM para modelagem de tópicos foi proposto pela primeira vez por Nigam *et al.* (2000) baseado na suposição de que cada documento é formado por apenas um tópico. Essa suposição é mais razoável para textos curtos do que a suposição de que cada texto é gerado por vários tópicos. Portanto, muitos modelos

para textos curtos foram propostos com base nessa hipótese.

Além do DMM, há extensões dessa técnica como o *Generalized Polya urn Poisson* DMM (GPU-PDMM), *Negative sampling and Quantization Topic Model* (NQTM), e *Laplacian* DMM (LapDMM).

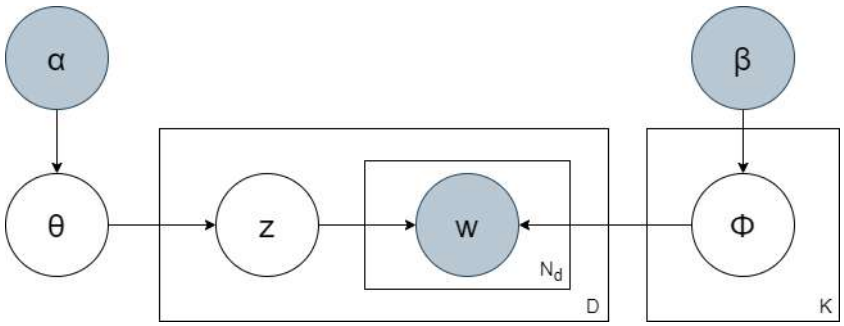
3.2.1.1 *Dirichlet Multinomial Mixture* (DMM)

O DMM é um modelo generativo que parte da premissa que cada documento abrange apenas um único tópico, pois, dado o conteúdo esparso dos textos curtos, essa suposição é mais razoável (ZHAO, W. X. *et al.*, 2011). Assim, essa suposição enriquece indiretamente as co-ocorrências de palavras no nível de documento, tornando o modelo mais eficaz para textos curtos.

O DMM é um modelo generativo que assume que as palavras em um documento são geradas a partir de uma distribuição multinomial, e os parâmetros desta distribuição multinomial são gerados a partir de uma distribuição Dirichlet. A distribuição Dirichlet é parametrizada por um vetor de concentração, que determina a probabilidade de cada tópico.

Formalmente, na modelagem de tópicos, Nigam *et al.* (2000) descreve que o DMM consiste em (1) K distribuições de tópico φ sobre o vocabulário de V palavras, extraídas de uma distribuição Dirichlet β e (2) uma distribuição ao nível de corpus θ sobre tópicos, extraída de um Dirichlet α anterior. Para cada documento d, ele primeiro desenha um indicador de tópico z_d de θ e, em seguida, desenha cada token de palavra w_{dn} do tópico selecionado φ_{zd}. A Figura 5 ilustra o processo do DMM. Esse processo também pode ser representado visualmente conforme a Figura 5.

Figura 5 – Representação gráfica do DMM.



Fonte: Adaptado de Yin e Wang (2014).

Dessa forma, dado um corpus de textos curtos D, o processo generativo do DMM pode ser descrito da seguinte forma:

1. Escolher uma distribuição por tópicos: $\theta \sim \textit{Dirichlet}(\alpha)$
2. Para cada tópico k

a) Fazer uma distribuição sobre palavras $\varphi_k \sim Dirichlet(\beta)$

3. Para cada documento d

   a) Fazer um tópico: $z_d \sim Multinomial(\theta)$

   b) Fazer cada uma das N_d palavras w_{dn}

      i. Fazer uma palavra: $w_{dn} \sim Multinomial(\varphi_{zd})$

Uma forma de evidenciar as variáveis ocultas no processo generativo é a amostragem de Gibbs. As variáveis ocultas referem-se às atribuições de tópicos dos documentos. Essas atribuições de tópicos não são diretamente observáveis no conjunto de dados e precisam ser inferidas pelo algoritmo. Nesse caso, a amostragem Gibbs é usada para evidenciar essas variáveis ocultas ao atualizar iterativamente as atribuições de tópicos dos documentos durante o processo generativo. No entanto, as atribuições de tópicos para cada documento são desconhecidas e precisam ser estimadas pelo algoritmo. A amostragem Gibbs entra em ação para atualizar essas atribuições de forma iterativa. Durante cada iteração, o algoritmo realiza a amostragem condicional das atribuições de tópicos, dado o modelo atualizado com base nas contagens de palavras por tópico e contagens de documentos por tópico. A amostragem Gibbs faz com que o algoritmo explore diferentes atribuições de tópicos dos documentos e converja para uma estimativa melhor dos tópicos no conjunto de dados. Através desse processo, as variáveis ocultas são evidenciadas e podem ser usadas para entender a estrutura subjacente dos tópicos presentes nos documentos.

Portanto, seguindo a abordagem de Yin e Wang (2014), um tópico z é amostrado para cada documento em cada iteração, seguindo a distribuição condicional:

$$P(w|z = k) = \frac{n_k^w + \beta}{\sum_v^V n_k^v + V\beta} \tag{6}$$

, onde n_k^w é a frequência de uma palavra w em um documento d. A probabilidade na equação (6) foi calculada usando estimativa pontual.

Uma característica que esse modelo e suas variantes mantêm, é a estrutura de tópico único para o documento. No entanto, um problema recorrente é que eles são sensíveis aos ruídos e palavras comuns, portanto, a representação de tópicos ao nível de documento pode ser calculada incorretamente (LI, X.; ZHANG, J.; OUYANG, 2019).

Esse problema é conhecido como problema de sensibilidade. Se a maioria das palavras em um texto curto não tem inclinação para o tópico ou mesmo ruídos, é provável que o tópico deste documento esteja mal calculado. Isso geralmente acontece porque os textos curtos contêm poucas palavras. Além disso, a suposição de que cada documento é gerado por um único tópico pode ser restritiva em alguns contextos, especialmente em casos onde o tamanho dos documentos se estendem mais que o comum e discutem vários tópicos.

Posteriormente, outros modelos surgiram baseados nas premissas do DMM, visando contornar os problemas inerentes do método.

3.2.1.2 *Generalized Polya urn Poisson* DMM (GPU-PDMM)

Chenliang Li *et al.* (2017) apresentam um algoritmo baseado no DMM, para solucionar duas limitações do DMM: primeiro, o DMM assume que cada texto, por ser curto, possui apenas um tópico. Isso pode ser problemático devido à subjetividade do que é considerado um texto curto e como o tamanho dos textos podem variar para cada conjunto de dados. Tendo isso em vista, a solução foi uma distribuição Poisson para determinar o número de tópicos. Assim, cada texto é associado a um pequeno número de tópicos (ex: um a três tópicos). Esse modelo foi chamado de PDMM.

O segundo problema do DMM é a falta de conhecimento prévio enquanto modelam textos curtos. As relações semânticas entre as palavras também são importantes para interpretar textos curtos. Com o auxílio de *word embeddings* é possível tratar dessa segunda limitação e aprender a relação semântica das palavras no corpus. Por meio do modelo *Generalized Polya urn* (GPU), o conhecimento sobre relações semânticas de palavras pode ser utilizado para classificar palavras relacionadas nos mesmos tópicos e melhorar a qualidade da modelagem de tópicos em textos curtos.

Ao estender o modelo PDMM com o modelo GPU, apresenta-se um novo modelo para modelagem de tópicos em textos curtos: o GPU-PDMM. Esse modelo destaca palavras semanticamente relevantes sob o mesmo tópico após a amostragem de um tópico num documento. Assim, forma-se uma conexão entre as palavras semanticamente relevantes, mesmo que compartilhem poucas ou nenhuma coocorrência no corpus.

Em suma, este modelo foi projetado para abordar as limitações do DMM padrão, principalmente a suposição de que cada documento é gerado por um único tópico. O GPU-DMM permite uma representação mais flexível, na qual cada documento pode ser atribuído a múltiplos tópicos, tornando-o mais adequado para textos que podem conter vários tópicos.

Partindo da premissa que um documento pode ser formado por um ou mais tópicos (mas não muitos), O GPU-DMM incorpora o modelo *Generalized Polya urn*: um modelo probabilístico que descreve a troca de bolas coloridas em uma urna.

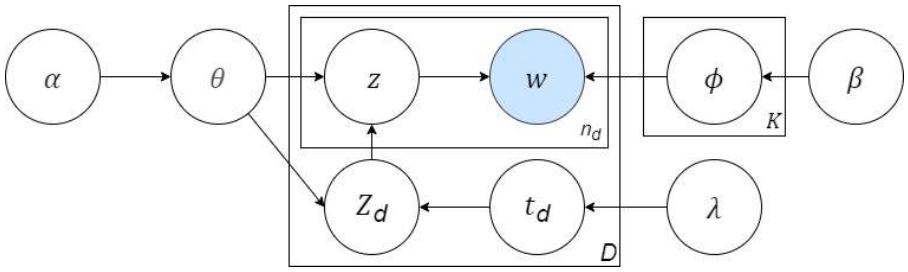
O *Polya urn* é um modelo estatístico que descreve um processo de amostragem com reposição (CHEN, M.-R.; KUBA, 2013). Os autores explicam que, supondo uma urna cheia de bolas de diferentes cores, inicialmente, a urna contém um número conhecido de bolas de cada cor. Em cada etapa do processo, uma bola é retirada da urna aleatoriamente, sua cor é registrada, e então ela é devolvida à urna com mais uma ou mais bolas da mesma cor. Isso significa que a probabilidade de escolher uma bola de uma determinada cor na próxima etapa aumenta cada vez que uma bola

dessa cor é escolhida. Portanto, este modelo é caracterizado por reforço positivo, onde cores escolhidas mais frequentemente tornam-se cada vez mais prováveis de serem escolhidas no futuro.

No modelo generalizado, uma urna contém bolas de diferentes cores, onde cada cor representa um tópico. Inicialmente, a urna tem uma configuração predefinida de bolas. Em cada etapa do processo, uma bola é sorteada aleatoriamente da urna e, em seguida, é devolvida à urna com um número predefinido de bolas da mesma cor. Isso leva a uma propriedade de "reforço" onde a probabilidade de sortear uma bola de uma determinada cor aumenta à medida que mais bolas dessa cor são adicionadas à urna. Portanto, no contexto da modelagem de tópicos, o modelo *Generalized Polya urn* permite que tópicos que já são proeminentes em um documento tenham maior probabilidade de serem reforçados, capturando assim a prevalência e a co-ocorrência de tópicos em documentos.

Logo, cada documento é gerado por $t_d (0 < t_d \leq \zeta)$ tópicos, onde ζ é o maior número de tópicos permitido em um documento. O GPU-PDMM usa a distribuição Poisson para modelar t_d. A Figura 6 mostra uma representação gráfica do GPU-PDMM.

Figura 6 – Representação gráfica do GPU-PDMM.



Fonte: Adaptado de Chenliang Li *et al.* (2017).

O processo generativo do GPU-PDMM pode ser descrito como:

1. Fazer uma proporção de tópico $\theta \sim Dirichlet(\alpha)$.
2. Para cada tópico $k \in 1, ..., K$:
    a) Fazer uma distribuição tópico-termo $\theta_k \sim Dirichlet(\beta)$
3. Para cada documento $d \in D$:
    a) Exemplo de um número de tópico $t_d \sim Poisson(\lambda)$.
    b) Exemplo t_d tópicos distintos $Z_d \sim Multinomial(\theta)$.
    c) Para cada palavra $w \in w_{d,1}, ..., w_{d,n_d}$:
        i. Exemplo uniforme de tópico $z_{d,w} \sim Z_d$
        ii. Exemplo de palavra $w \sim Multinomial(\varphi_{z_{d,w}})$.

Aqui os exemplos t_d são escolhidos usando distribuição Poisson com parâmetro λ e Z_d é o conjunto de tópicos para o documento d. Devido aos custos computacionais em realizar amostragem de Z_d, o GPU-PDMM infere a probabilidade de tópicos $p(z|d)$ de cada documento d conforme a equação:

$$p(z = k|d) \propto \sum_{w \in d} p(z = k|w)p(w|d) \tag{7}$$

, onde $p(w|d) = \frac{n_d^w}{n_d}$. Consequentemente, o GPU-PDMM apenas escolhe os M maiores tópicos para os d documentos baseado na probabilidade $p(z|d)$ para gerar Z_d, onde $\zeta < M \leq K$.

No entanto, assim como outros modelos de tópicos, o GPU-DMM tem suas limitações. A principal é a complexidade computacional, pois o modelo requer a execução de um algoritmo de amostragem, que pode ser intensivo em termos de tempo e recursos para grandes conjuntos de dados.

Apesar disso, experimentos conduzidos em duas coleções de texto reais, em duas línguas diferentes (inglês e chinês), mostram que o GPU-PDMM alcança uma melhor representação de tópicos que modelos do estado-da-arte.

3.2.1.3 *Negative sampling and Quantization Topic Model* (NQTM)

O NQTM é um método de modelagem de tópicos para texto curto que pode ser treinado com métodos de *gradient descent*, assim como outros modelos de *machine learning*. NQTM é uma estrutura de codificação automática que aborda o problema de modelagem de texto curto não supervisionado (WU, X. *et al.*, 2020). Neste modelo, implementa-se duas etapas essenciais: mapear a distribuição de tópicos em um espaço de embedding apropriado; e um decodificador de amostragem negativa para melhorar a diversidade dos tópicos.

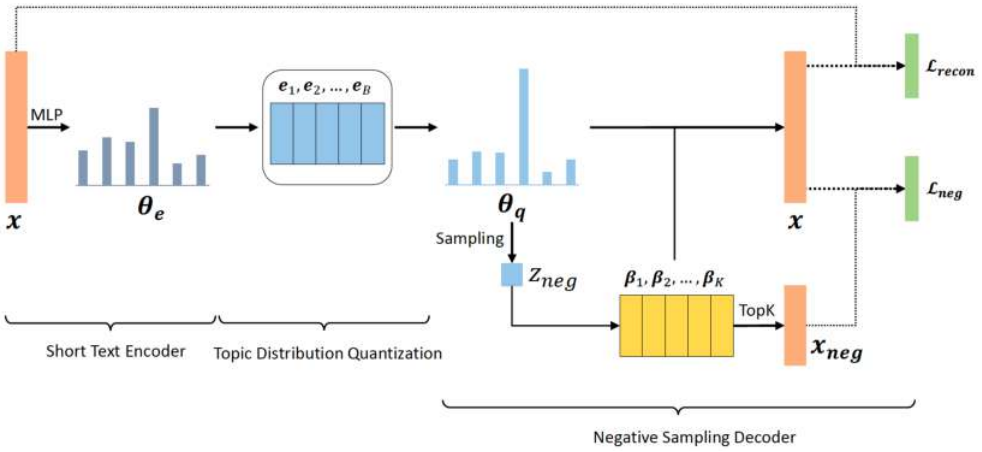
A amostragem negativa é uma técnica que permite ao modelo aprender a distinção entre tópicos relevantes e irrelevantes para um documento. Em vez de considerar todas as palavras em todos os tópicos, o NQTM seleciona um pequeno número de exemplos negativos (palavras que não estão no tópico atual) durante o treinamento. Isso reduz a complexidade computacional e melhora a eficiência do modelo. A quantização, por outro lado, é usada para limitar o número de tópicos que uma palavra pode ter. Isso simplifica a representação do modelo e melhora sua interpretabilidade.

O NQTM também emprega uma arquitetura de rede neural para modelar a distribuição de tópicos e palavras, o que permite ao modelo capturar relações não lineares e complexas nos dados. Para isso, os Perceptrons multicamada (MLPs) são usados para o *encoder* e *decoder*. Segundo Popescu *et al.* (2009), os MLPs são uma classe de redes neurais artificiais *feedforward* que consistem em múltiplas camadas de neurônios, ou nós, cada uma realizando cálculos e transformações lineares e não-

lineares nos dados. Eles são treinados usando um algoritmo chamado retropropagação, que ajusta os pesos da rede de maneira iterativa para minimizar a diferença entre a saída prevista da rede e a saída desejada, permitindo ao MLP aprender e modelar relações complexas e não-lineares entre as entradas e as saídas.

Além disso, o modelo necessita que o texto curto *x* esteja no formato de *bag-of-words*. Conforme mostra a Figura 7, o modelo consiste em três etapas principais.

Figura 7 – Arquitetura do NQTM.



Fonte: Xiaobao Wu *et al.* (2020).

A Figura 7 mostra como o NQTM foi estruturado em três etapas: *enconder* de texto curto; quantização da distribuição de tópicos; e *decoder* de amostragem negativa.

Durante a etapa do *enconder* de textos curtos, na adoção de uma estrutura de rede simples, θ_e é calculado da seguinte forma: $\pi_1 = \zeta(W_1 x + b_1)$ e $\pi_2 = \zeta(W_2 \pi_1 + b_2)$, para então calcular $\theta_e = \sigma(\pi_2)$.

Após o *encoding*, começa a etapa de quantização da distribuição de tópicos. Define-se um espaço de *embedding* discreto $e = (e_1, e_2, ..., e_B) \in R^{K \times B}$, sendo K o número de tópicos e B é o tamanho do espaço. A matriz de incorporação é estendida para atingir o máximo da distância entre os vetores de *embedding*, os primeiros K vetores são inicializados com uma matriz identidade e um conjunto adicional, inicializados com escala de unidade uniforme.

A representação da matriz *embedding* é otimizada durante o treinamento. O texto curto x é codificado para o θ_e e então mapeado para o espaço *embedding* como $\theta_q = e_k$, onde $k = \arg\min_j \left\| \theta_e - e_j \right\|_2$. θ_φ é alterado para o e_k que mantém a distância mais próxima. Então o novo θ_φ pode ser passado para as etapas do decodificador, que também são MLPs.

Enfim, um esquema de amostragem negativa é formulado para gerar tópicos mais diversos. As palavras com alta probabilidade em outros tópicos, mas não atri-

buídas ao documento atual, são chamadas amostras negativas. O modelo escolhe os maiores tópicos t e os remove, que se tornam os tópicos negativos Z_n com igual probabilidade. Então, gera-se M palavras a partir de Z_n, pois o *decoder* não inclui as palavras dos principais tópicos t.

Nesse processo, a função de perda tem é formada pelos componentes de t e as amostras negativas:

$$L_{recon}(x^{(i)}) = -x^{(i)} \log(\sigma(\beta\theta_q^{(i)})) \tag{8}$$

$$L_{neg}(x^{(i)}) = -x_{neg}^{(i)} \log(1 - \sigma(\beta\theta_q^{(i)})) \tag{9}$$

onde $x^{(i)}$ refere-se ao i-ésimo documento do corpus. A forma final da função de perda é dada com uma regularização.

Com isso, o objetivo é obter a coleção de tópicos e também os tópicos para cada um dos textos curtos de entrada. Com o espaço de embedding, o NQTM primeiro obtém o t para cada documento e, em seguida, extrai as palavras de tópico associadas a ele.

Apesar disso, como muitos modelos baseados em redes neurais, o NQTM pode ser mais difícil de interpretar e requer mais recursos computacionais para treinamento e inferência em comparação com modelos de tópicos tradicionais baseados em probabilidades. No entanto, sua capacidade de lidar com grandes conjuntos de dados e fornecer representações de tópicos mais precisas torna o NQTM uma adição importante ao campo de algoritmos de modelagem de tópicos.

3.2.1.4 *Laplacian* DMM (LapDMM)

A ideia básica do LapDMM é estender o DMM enquanto preserva a estrutura de vizinhança local de textos curtos, permitindo espalhar sinais de tópicos entre documentos vizinhos, para sanar o problema de sensibilidade (LI, X. *et al.*, 2021). Esse problema é contornado com regularização Manifold. Uma vez que os dados residem em um subespaço de menor dimensão (ou seja, um Manifold) no espaço de entrada de alta dimensão, propõe-se que essa estrutura seja explorada para melhorar a modelagem e a predição.

No contexto de modelagem de tópicos, a regularização Manifold delimita as representações de tópicos latentes a pares de documentos. Eles são semelhantes entre si se forem vizinhos mais próximos. Formalmente, é representado como um grafo direcionado de documentos com D vértices, onde cada vértice corresponde a um documento do corpus, e W representa a matriz de pesos das arestas.

Logo, no LapDMM a regularização Manifold é implementada através da utilização da matriz Laplaciana, que captura a estrutura de Manifold dos dados. Ou seja, a matriz Laplaciana fornece uma representação da estrutura de vizinhança dos da-

dos, onde textos curtos que são "vizinhos" próximos no Manifold são provavelmente do mesmo tópico. Portanto, ao integrar essa regularização Manifold no modelo, o LapDMM incorpora informações contextuais adicionais que ajudam a melhorar a precisão e a interpretabilidade dos tópicos descobertos.

Apesar da solução para o problema de sensibilidade, há um obstáculo: a regularização de Manifold não pode ser diretamente incorporada na inferência do DMM. Devido ao DMM restringir apenas um tópico para cada documento, não há representações claras de tópicos K-dimensional para documentos. Para isso, foi utilizado um regularizador Manifold em relação à distribuição variacional.

Graças ao design do DMM, as duas distribuições φ e θ podem ser marginalizadas diretamente. Então define-se uma distribuição variacional *mean-field* em relação à atribuição de z tópicos de documentos:

$$q(z|\gamma) = \prod_{d=1}^{D} q(z_d|\gamma_d) \tag{10}$$

, onde cada $q(z_d|\gamma_d)$ é uma distribuição multinomial com um vetor K-dimensional de parâmetro variacional γ_d. Dado um corpus S formado por textos curtos, treina-se o DMM maximizando o seguinte objetivo variacional em relação a γ:

$$L(\gamma) = \mathbb{E}_{q(z|\gamma)}[\log p(S,z|\alpha,\beta) - \log q(z|\gamma)] \tag{11}$$

Define-se um regularizador Manifold para $q(z)$, visto que cada distribuição variacional do documento $q(z_d|\gamma_d)$ é usado uma aproximação do tópico do documento atual. Com isso tem-se a equação da regularização Manifold, representada pela equação:

$$R(\gamma) = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{D}(\gamma_{ik} - \gamma_{jk})^2 W_{ij} \tag{12}$$

Assim, ao combinar as Equações (2) e (3), tem-se a equação com o LapDMM em relação a γ:

$$\mathcal{L}(\gamma) = \frac{1}{D}L(\gamma) - \lambda R(\gamma) \tag{13}$$

onde $\lambda \in [0,1]$ é um parâmetro de regularização.

Dessa forma atinge-se a formulação Manifold do LapDMM, descrita pela distância euclidiana entre parâmetros variacionais. Os testes mostraram que essa forma é mais fácil de calcular do que outras.

Todavia, o aumento da complexidade computacional introduzido pela utilização da matriz Laplaciana e regularização Manifold pode ser um desafio ao lidar com conjuntos de dados muito grandes. Mas apesar disso, o LapDMM representa um avanço significativo na modelagem de tópicos de textos curtos, oferecendo uma abordagem mais robusta e contextualmente enriquecida para a descoberta de tópicos.

3.2.2 Baseados em auto-agregação

Muitos algoritmos que dependem da co-ocorrência de palavras em um documento perdem consideravelmente em performance, classificação e qualidade dos tópicos gerados devido ao curto tamanho dos textos (LI, Chenliang *et al.*, 2017).

Nesse cenário surgiram os métodos baseados em autoagregação, onde os textos curtos são agregados a longos pseudo-documentos antes da inferência de tópicos, resolvendo a questão da baixa co-ocorrência de palavras num documento (GAO, W. *et al.*, 2019). O sucesso da modelagem de tópicos nas mídias sociais por meio da agregação heurística de tuítes mostrou-se uma solução para a limitação de contexto dos documentos. Com isso, pode-se recorrer a algoritmos de agrupamento automático para agregar textos curtos em pseudodocumentos longos, antes que um modelo de tópico padrão seja aplicado.

Alguns modelos de destaque nessa área são: *Self-Aggregation Based Topic Model* (SATM), *Conditional Random Field Regularized Topic Model* (CRFTM), *Pseudo-Document-Based Topic Modeling* (PTM), e *Meta-Info Guided Aggregation* (MIGA).

3.2.2.1 Self-Aggregation based topic model (SATM)

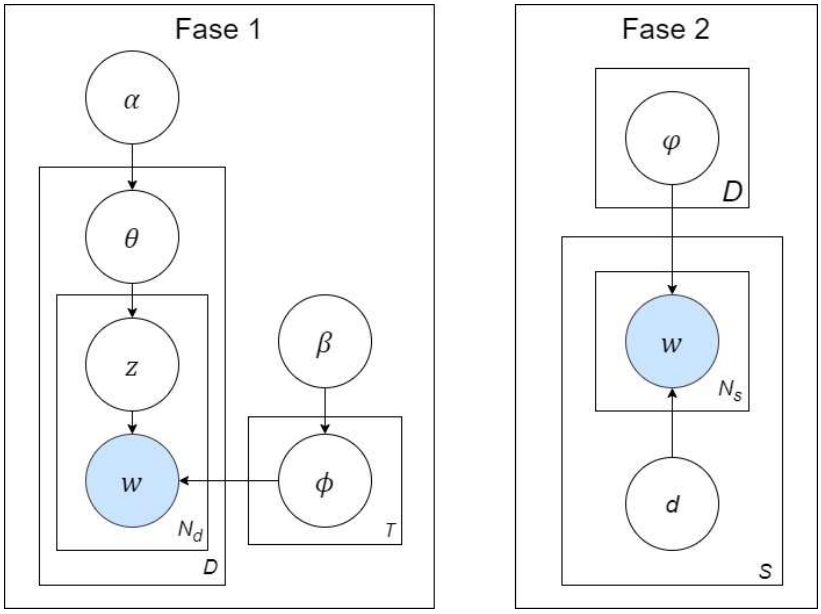
A abordagem de agregar pseudo-documentos longos demonstrou ser eficiente para contornar os problemas de esparsidade. Muitas soluções como essa são documentadas em pesquisas sobre mídias sociais, como o Twitter, e são, geralmente, agregados por alguma métrica observável como autor, hashtags ou localização (HONG, L.; DAVISON, 2010).

O desenvolvimento de métodos de agregação generalizados para textos curtos foi observado mais recentemente e supera o obstáculo do contexto limitado. A abordagem *Self-Aggregation Based Topic Model* é uma delas (QUAN *et al.*, 2015).

A ideia central do SATM é que textos curtos relacionados ao mesmo tópico tendem a se agrupar. Assim, em vez de tratar cada texto curto como um documento independente, o SATM agrega textos curtos similares em grupos maiores, ou “documentos longos”, antes de aplicar a modelagem de tópicos. Este processo de autoagregação permite que o SATM capture melhor a semântica dos textos curtos, melhorando a qualidade dos tópicos descobertos.

Os autores descrevem o processo gerador do SATM em duas fases: a primeira fase segue a hipótese de modelos de tópicos padrão, como o LDA, para gerar um conjunto de D documentos de tamanho regular, onde cada documento d é composto por uma sequência de palavras, w_d, de tamanho N_d. Isso significa que para cada trecho de texto s contendo uma sequência v_s, de N_s palavras, existe exatamente um documento longo responsável por sua geração. Todo esse procedimento generativo pode ser descrito conforme a Figura 8.

Figura 8 – Representação gráfica das fases 1 e 2 do SATM, respectivamente.



Fonte: Adaptado de Quan *et al.* (2015).

Os autores descrevem o processo do SATM como:

1. Para cada tópico latente z:
    a) Fazer uma distribuição multinomial sobre as palavras $\varphi_z \sim Dirichlet(\beta)$.
2. Exemplo de uma proporção de tópico θ_d $Dirichlet(\alpha)$.
3. Para cada palavra $w \in w_d$:
    a) Exemplo de um tópico $z_w \in Multinomial(\theta_d)$.
    b) Exemplo de uma palavra $w \in Multinomial(\Phi_{z_w})$.
4. Para cada palavra $w \in v_s$:
    a) Selecionar uma palavra φ_d, a distribuição de probabilidade sobre palavras para o documento d.

onde α e β são hiperparâmetros, e Φ_{z_w} refere-se à distribuição multinomial sobre palavras para o tópico z_w. Os passos 1 a 3 do processo correspondem à primeira fase da geração dos documentos longos, e o passo 4 corresponde à segunda fase.

O SATM também envolve d e φ como novas variáveis ocultas, além das mesmas variáveis ocultas de z, φ, e θ como nos modelos tradicionais. Consequentemente, o principal problema na modelagem de tópicos do SATM é estimar a distribuição posterior das variáveis ocultas θ_d, d_s, e z_s para um determinado trecho de texto curto, que equivale a:

$$p(\theta_d, d_s, z_s | v_s, \alpha, \beta) = \frac{p(\theta_d, d_s, z_s, v_s | \alpha, \beta)}{p(v_s | \alpha, \beta)} \tag{14}$$

onde z_s representa a sequência de identidades de tópicos designadas para as palavras de s, e d_s é o documento oculto para gerar s.

Apesar do bom desempenho demonstrado pelo modelo, existem muitos contratempos para essa abordagem. Esse modelo emprega variáveis latentes adicionais não observadas, que devem ser consideradas na otimização: o número de documentos longos. Além disso, o processo de autoagregação pode ser sensível à escolha da medida de similaridade usada para agrupar os textos curtos. O total de variáveis no SATM aumenta à medida que conjuntos de dados maiores são usados, tornando o modelo suscetível a *overfitting* e resultando em uma complexidade de tempo inviável para uso prático.

3.2.2.2 *Conditional Random Field Regularized Topic Model* (CRFTM)

Wang Gao *et al.* (2019) apresentam o CRFTM como uma abordagem de modelagem de tópicos para textos curtos para enfrentar o problema da esparsidade de dados em textos curtos. Segundo os autores, o CRFTM visa solucionar duas questões: (1) encontrar uma solução com boa generalização para agregar textos curtos, mitigando a esparsidade de dados; (2) como diferenciar as sutilezas no sentido das palavras entre os tópicos, com auxílio de *word embeddings* para melhorar a modelagem de tópicos em textos curtos. O CRFTM introduz uma regularização baseada em *Conditional Random Fields* (CRF), uma estrutura probabilística usada para modelar sequências de variáveis.

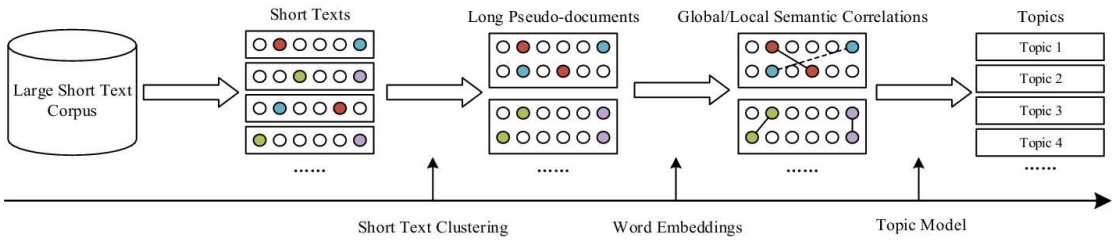
A regularização baseada em CRF é projetada para capturar a dependência entre palavras adjacentes nos textos. Em contraste com modelos de tópicos tradicionais que tratam as palavras como independentes, o CRFTM reconhece que as palavras em um texto curto são frequentemente semanticamente inter-relacionadas. O modelo, portanto, usa CRFs para modelar a dependência entre palavras vizinhas, permitindo uma melhor detecção de tópicos em textos curtos.

Para tanto, o CRFTM adota uma estrutura de duas fases para abordar problemas de dispersão e desambiguação no sentido das palavras na modelagem de tópicos em textos curtos. A Figura 9 mostra as duas fases do modelo proposto.

Na primeira fase, o CRFTM agrega textos curtos em pseudodocumentos longos usando uma métrica própria que mede a distância entre textos curtos. Essa métrica, chamada de *Embedding-Based Minimum Average Distance* (EMAD), captura diretamente pares de palavras semanticamente relacionadas em dois textos distintos. Esses pares de palavras tem maior probabilidade de pertencer ao mesmo tópico.

Na segunda fase, define-se um CRF na camada do LDA para aumentar a coerência da modelagem de tópicos. O modelo define dois tipos de correlações: (1) correlações semânticas globais são usadas para encorajar palavras relacionadas a compartilhar o mesmo tópico, o que pode melhorar a coerência dos tópicos aprendi-

Figura 9 – Visão geral do modelo CRFTM



Fonte: Wang Gao *et al.* (2019).

dos; (2) CRFTM utiliza correlações semânticas locais para identificar efetivamente os sentidos de palavras ambíguas, assim, as palavras irrelevantes podem ser filtradas.

Em geral, cada texto curto é representado como uma concatenação de palavras. Dadas *embeddings* de palavras pré-treinadas de cada palavra, mede-se a distância entre as palavras pela distância do cosseno entre suas representações vetoriais. A distância entre as palavras w_a e w_b é definida como:

$$d(w_a, w_b) = 1 - \frac{w_a w_b}{\|w_a\| \, \|w_b\|} \tag{15}$$

Sejam os vetores v_i e v_j a representação de dois textos curtos contendo U e R palavras, respectivamente. Primeiro, seja $\mathbb{T} \in \mathbb{R}^{U \times R}$ uma matriz de distâncias onde $\mathbb{T}_{u,r}$ denota a distância entre a palavra u em v_i e a palavra r em v_j. Além disso, calcula-se a média do valor mínimo de cada linha de $\mathbb{T}$ para representar o EMAD de v_i para v_j, ou seja, $d(v_j||v_i) = \frac{1}{R} \sum_r \min(\mathbb{T}_r)$.

O EMAD do texto curto s_i contendo palavras de outros textos curtos é frequentemente maior do que o EMAD de outras sentenças de s_i. Portanto, criou-se uma medida de distância assimétrica definindo $d(v_i, v_j) = \min(d(v_i||v_j), d(v_j||v_i))$ para capturar mais pares de palavras semanticamente relacionadas.

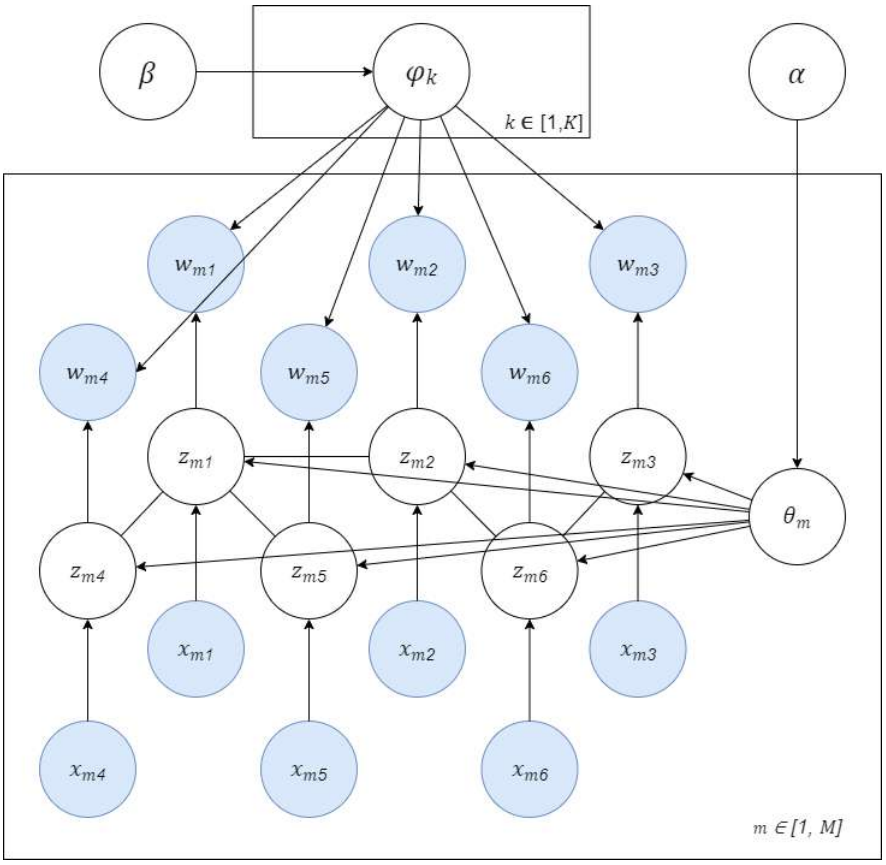
Após obter a distância entre os textos curtos, estes são agregados em pseudodocumentos longos com base no algoritmo de clustering $k-medoids$ (KAUFMAN; ROUSSEEUW, 1990). O algoritmo $k-medoids$ é baseado em técnicas de objetos representativos onde os centroides são substituídos por medoids para representar clusters, sendo o medoid o objeto de dados mais ao centro de um cluster. Este clustering é executado iterativamente, de forma que os medoids mudam sua localização passo a passo até que nenhuma mudança aconteça. Assim, após o $k-medoids$, todos os textos são agrupados em M pseudodocumentos longos.

Com isso, as palavras com correlação semântica mais alta são agrupadas no mesmo tópico (GAO, W. *et al.*, 2019). Para fazer isso, mede-se a distância entre duas palavras pela distância de cosseno no espaço de *embedding*. Se a distância entre duas palavras num pseudodocumento for menor que um limite, significa que elas têm uma

correlação semântica global entre si e é provável que pertençam ao mesmo tópico.

O CRF é um modelo gráfico probabilístico usado para codificar várias relações conhecidas entre observações (LAFFERTY; MCCALLUM; PEREIRA, 2001). Com base nisso, o CRFTM define um CRF sobre a camada de tópico latente. Dado um pseudo-documento m contendo N_m palavras $\{w_{mi}\}_{i=1}^{N_m}$, considera-se cada par de palavras (w_{mi}, w_{mj}). Caso eles sejam semanticamente correlatos, ou seja, d(w_{mi},w_{mj}) < μ, cria-se uma aresta não direcionada entre tópicos (z_{mi},z_{mj}). A Figura 10 mostra a relação entre os parâmetros.

Figura 10 – Representação gráfica do CRFTM.



Fonte: Adaptado de Wang Gao *et al.* (2019).

Portanto, o processo generativo pode ser resumido da seguinte forma:

1. Fazer uma proporção de tópico $\Theta \sim Dir(\alpha)$.
2. Para cada tópico k:
    a) Fazer uma proporção $\varphi_k \sim Dir(\beta)$.
3. Para cada pseudodocumento m.
    a) Fazer uma atribuição de tópico z_m
    b) Obter a palavra observada $w_{mi} \sim Mult(\varphi_{z_{mi}})$

Dados os rótulos dos tópicos, a geração de palavras é a mesma do LDA. w_{mi} é gerado a partir da distribuição multinomial de palavras-tópico $\varphi_{z_{mi}}$ correspondente a z_{mi}.

Como mostra a Figura 10, o CRFTM estende o modelo LDA impondo um CRF na camada de tópico latente para incorporar correlações semânticas globais e locais no momento de atribuição dos tópicos. Dentre as variáveis, M denota o número de pseudodocumentos, e K refere-se ao número de tópicos latentes. Cada pseudodocumento m possui N_m palavras. w_{mn} é o valor observável da enésima palavra em m. α e β e são hiperparâmetros, sendo *alpha* a força relativa de tópicos latentes ocultos no conjunto de pseudodocumentos, e β é a probabilidade de distribuição de todos os tópicos ocultos. θ_m denota a distribuição de probabilidade de tópico para um pseudodocumento m. φ_k representa a distribuição de palavras para determinado tópico k. z_{mn} refere-se ao rótulo do tópico atribuído à enésima palavra no pseudodocumento m. E x_{mn} representa a palavra contextual do documento, usada para identificar a correlação semântica entre diferentes palavras num pseudodocumento.

No entanto, como outros modelos de tópicos, o CRFTM tem suas limitações. A principal é a complexidade computacional: a inclusão da regularização CRF aumenta a complexidade do modelo, tornando-o mais exigente em termos de recursos computacionais. Além disso, a escolha do número de tópicos a priori continua sendo um desafio. Apesar dessas limitações, o CRFTM representa um avanço significativo na modelagem de tópicos, proporcionando uma representação mais precisa dos tópicos em documentos, especialmente no caso de textos curtos.

3.2.2.3 *Pseudo-document-Based Topic Modeling* (PTM)

Zuo *et al.* (2016) apresentam o PTM. Este método, supõe que cada texto curto é uma amostra de um pseudodocumento longo p_l, e então infere os tópicos latentes do conjunto de pseudodocumentos P.

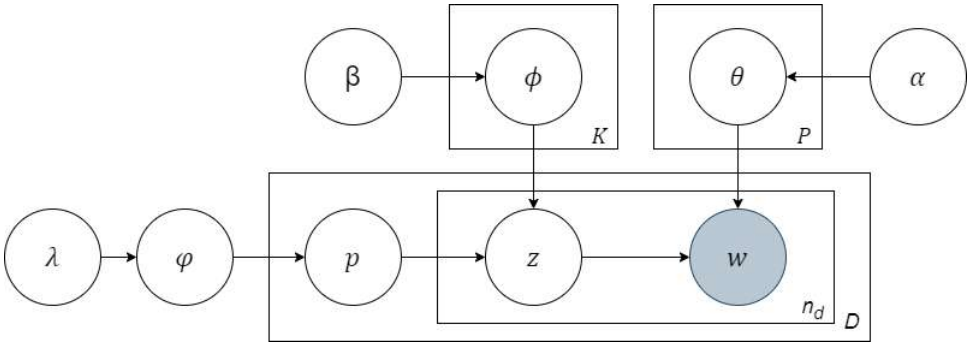
A ideia por trás do PTM difere dos outros modelos de tópicos. Em vez de tratar cada texto curto como um documento independente, o PTM agrega textos curtos semelhantes em pseudo-documentos mais extensos. Em seguida, um modelo de tópicos tradicional, como o LDA, pode ser aplicado aos pseudo-documentos para descobrir os tópicos. Ao agrupar os textos curtos dessa maneira, o PTM pode superar a escassez de palavras em textos curtos e melhorar a qualidade dos tópicos descobertos.

Uma distribuição multinomial φ é usada para modelar a distribuição de textos curtos sobre pseudodocumentos. Um modelo gráfico do PTM é mostrado na Figura 11.

O processo generativo do PTM pode ser descrito como:

1. Amostragem de $\varphi \sim \textit{Dirichlet}(\lambda)$.
2. Para cada tópico $k \in [1, K]$:

Figura 11 – Representação gráfica do modelo PTM.



Fonte: Adaptado de Zuo *et al.* (2016).

a) Obter $\varphi_k \sim Dirichlet(\beta)$.

3. Para cada pseudodocumento l:

   a) Obter amostra de $\theta_l \sim Dirichlet(\beta)$

4. Para cada documento $d \in D$:

   a) Obter amostra de pseudodocumento $l \sim Multinomial(\varphi)$.

   b) Para cada palavra $ww_{d,1}, ..., w_{d,n_d}$ em d:

      i. Obter amostra de tópico $z \sim Multinomial(\theta_l)$.

      ii. Obter amostra de palavra $w \sim Multinomial(\varphi_z)$.

Integrando θ, φ, e φ, à atribuição de pseudodocumento l para textos curtos d se dão podem ser estimados como:

$$p(l_d = l|P_{\neg d}, D) \propto \frac{m_{l,\neg d}}{N-1+\lambda\left|P\right|} \frac{\prod_{k\in d}\prod_{j=1}^{n_d^k}(n_{d,\neg d}^k+\alpha+j-1)}{\prod_{i=1}^{n_d}(n_{l,\neg d}+K\alpha+i-1)} \tag{16}$$

onde m_l é o número de textos curtos associados ao pseudodocumento l, sendo n_l^k o número de palavras associadas com o tópico k em l. Após obter os pseudodocumentos de cada texto curto, o PTM obtém amostras das atribuições de tópicos para cada palavra w no documento d, isto é,

$$p(z_{d,w} = k|Z_{\neg(d,w)}, D) \propto (n_l^k+\alpha)\frac{n_k^w\beta}{n_k+V\beta} \tag{17}$$

Assim, calcula-se a distribuição palavra-documento θ_d como,

$$\theta_d^k = \frac{n_d^k+\alpha}{n_d+K\alpha} \tag{18}$$

Todavia, o modelo PTM também enfrenta algumas limitações. A escolha da medida de similaridade usada para agrupar os textos curtos pode afetar significativamente

os resultados. Apesar disso, o PTM representa um avanço importante na modelagem de tópicos de textos curtos, proporcionando uma estratégia eficaz para melhorar a detecção de tópicos em tais contextos.

3.2.2.4 *Meta-Info Guided Aggregation* (MIGA) *model*

He Zhao *et al.* (2019) apresentam o modelo MIGA como uma forma de incorporar a meta informação, ou metadados, de cada documento diretamente no processo generativo do modelo de autoagregação, em um modelo único e integrado. O MIGA agrega os textos curtos de acordo com dois fatores: similaridade de conteúdo e similaridade de metadados. Dessa forma, o modelo faz um balanço matemático de forma que quanto mais os documentos compartilham metadados e discutem tópicos similares, mais provável seja que eles pertençam ao mesmo cluster.

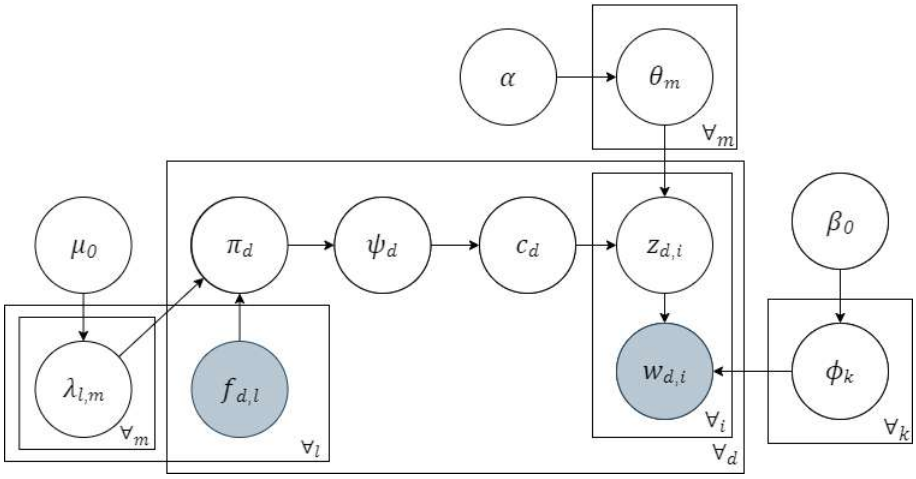
A ideia fundamental do MIGA é que textos curtos com meta-informações semelhantes são prováveis de pertencer ao mesmo tópico. Portanto, ao invés de tratar cada texto curto como um documento independente, o MIGA agrupa textos curtos com base na semelhança das suas meta-informações para formar pseudo-documentos. Posteriormente, os modelos de tópicos tradicionais podem ser aplicados a estes pseudo-documentos para descobrir os tópicos. Ao incorporar meta-informações no processo de agregação, o MIGA consegue melhorar a qualidade dos tópicos descobertos.

O MIGA se difere de uma fórmula comumente adotada entre modelos de autoagregação, que impõem uma priori não informativa em ψ_d, enquanto o MIGA parte de uma priori $\pi_d \in \mathbb{R}_+^M$ específica de documento, construído a partir da meta informação de d. Dado um conjunto de L metadados únicos em um corpus, os dados do documento d são codificados em um vetor binário $f_d \in \{0,1\}^L$, onde $f_{d,l} = 1$ indica que d possui o metadado l. Essa forma de codificação permite que um documento possua múltiplos metadados. A Figura 12 mostra o processo generativo do MIGA.

O processo generativo do MIGA pode ser descrito como:

1. Para cada cluster latente m:
    a) Para cada metadado l: $\lambda_{l,m} \sim Ga(\mu_0,\mu_0)$
    b) Obter: θ_m $Dir_K(\alpha)$
2. Para cada tópico k, obter φ_k $Dir_V(\beta_0)$
3. Para cada documento d:
    a) Para cada cluster m, computa $\pi_{d,m} = \prod_{l=1}^{L}(\lambda_{l,m})^{f_{d,l}}$
    b) Obter: $\psi_d \sim Dir_M(\pi_d)$
    c) Obter a atribuição de cluster: $c_d \sim Cat_M(\psi_d)$
4. Para cada palavra i no documento d:
    a) Obter tópico: $z_{d,i} \sim Cat_K(\theta_{c_d})$

Figura 12 – Representação gráfica do modelo MIGA.



Fonte: Adaptado de He Zhao *et al.* (2019).

b) Obter palavra: $w_{d,i} \sim Cat_V(\varphi_{z_{d,i}})$

onde $Ga()$ é a distribuição gama com os parâmetros de forma e taxa; $Dir_K()$ é a distribuição Dirichlet dimensional K; $Cat_K()$ é a distribuição categórica dimensional K.

A ideia principal do MIGA é a agregação guiada de metadados, onde, em vez de colocar um priori não informativo em psi_d, constrói-se um Dirichlet informativo específico de documento com o parâmetro π_d computado dos metadados do documento. Portanto, $\lambda_{l,m}$ captura as correlações entre o metadado l e o cluster m. Então, se $f_{d,l} = 1$, $\lambda_{l,m}$ contribui para $\pi_{d,m}$, o priori de $\psi_{d,m}$. Dessa forma, pode-se observar como a meta informação influencia a probabilidade de um documento ser designado a um tópico.

Além disso, o modelo pode ser estendido para incorporar *embeddings* de palavras para guiar a geração de tópicos latentes. Seguindo a abordagem de He Zhao *et al.* (2017), obtém-se $\varphi_k \sim Dir_V(\beta_k)$, onde $\beta_k \in \mathbb{R}_+^V$ é computado com um modelo log-linear de *embeddings* de palavra, similar ao passo 3a do processo generativo do MIGA.

No entanto, apesar das vantagens oferecidas pelo MIGA, o modelo também enfrenta alguns desafios. A seleção e a interpretação de meta-informações relevantes podem ser complexas e, além disso, a decisão sobre o número ideal de tópicos a priori continua sendo uma questão desafiadora entre os algoritmos. Apesar dessas limitações, o MIGA representa um avanço significativo na modelagem de tópicos para textos curtos, oferecendo uma abordagem mais robusta e contextualmente enriquecida para a descoberta de tópicos.

3.2.3 Baseados em co-ocorrência global

Alguns modelos buscam superar a esparsidade dos dados locais ao considerar padrões globais (no corpus) de co-ocorrência de palavras, desta forma a análise da relação semântica das palavras não se limita apenas a uma unidade de documento do corpus (RASHID; SHAH; IRTAZA, 2019). Há dois tipos de modelos nessa categoria (QIANG *et al.*, 2020): o primeiro tipo usa diretamente as co-ocorrências de palavras globais para a inferência de tópicos, como o BTM (CHENG *et al.*, 2014). O segundo tipo constrói redes de co-ocorrência de palavras e então infere os tópicos a partir dessas redes. Representados em grafos, cada palavra corresponde a um nó e o peso das arestas corresponde à probabilidade de co-ocorrência entre as duas palavras conectadas (ZUO *et al.*, 2016).

Modelos como o *Biterm Topic Modeling* (BTM), *Word Network Topic Model* (WNTM), *Multiterm Topic Model* (MTM), e *Noise Biterm Topic Model with Word Embeddings* (NBTMWE).

3.2.3.1 *Biterm topic modeling* (BTM)

Apresentado por Cheng *et al.* (2014), o BTM se baseia na ideia de *biterms*, onde se identificam os padrões de co-ocorrência de palavras em todo o corpus para resolver o problema de esparsidade de co-ocorrências ao nível de documento. Dessa forma, qualquer par de palavras de apareça em um documento é tratado como um *biterm*.

O BTM aborda algumas das principais limitações dos modelos de tópicos tradicionais, como o LDA, que presume que cada documento contém vários tópicos. Esta premissa é problemática para textos curtos, que geralmente contêm apenas um ou dois tópicos devido à sua natureza concisa. O BTM, ao contrário, modela diretamente as relações entre os *biterms*, em vez de documentar o nível de misturas de tópicos.

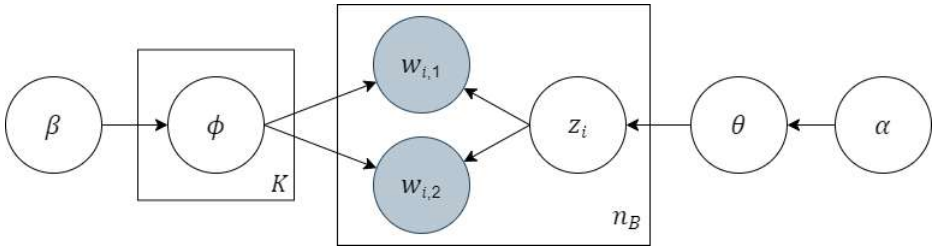
Em vez de tratar cada texto curto como um documento independente, o BTM considera o conjunto completo de textos como um único documento e modela a co-ocorrência de pares de palavras em todo o conjunto de textos. Cada par de palavras que ocorre no mesmo intervalo é tratado como um *biterm*. O modelo então infere a distribuição de tópicos para cada *biterm*, permitindo a descoberta de tópicos mais precisos em textos curtos.

Primeiramente, o BTM gera os *biterms B* do corpus *D*, para inferir os tópicos com base em *B*. Os *biterms* de um corpus que contenha n_b *biterms* são representados como $B = \{b_i\}_{i=1}^{n_B}$, onde $b_i = (w_{i,1}, w_{i,2})$. Um modelo gráfico do BTM é ilustrado na Figura 13, seguido de uma descrição de seu processo generativo.

O processo generativo do BTM é descrito como:

1. Obter $\theta \sim$ *Dirichlet*(α).
2. Para cada tópico $k \in [1, K]$:

Figura 13 – Representação gráfica do modelo BTM.



Fonte: Adaptado de Cheng *et al.* (2014).

a) Obter $\psi_k \sim Dirichlet(\beta)$

3. Para cada biterm $b_i \in B$:

   a) Obter $z_i \sim Multinomial(\theta)$,

   b) Obter $w_{i,1}, w_{i,2} \sim Multinomial(\psi_{zi})$.

No BTM, o vetor de características do tópico k é definido com a tupla $\{n_k(w \in W), n_k\}$. As propriedades dessas características são descritas a seguir:

1. **Propriedades adicionáveis**. Um *biterm* b_i pode ser eficientemente adicionado ao tópico k ao atualizar seu vetor de característica, de forma que:

$$n_k^{w_{i,1}} = n_k^{w_{i,1}} + 1; n_k^{w_{i,2}} = n_k^{w_{i,2}} + 1; n_k = n_k + 1 \tag{19}$$

2. **Propriedades deletáveis**. Um *biterm* b_i pode ser eficientemente deletado do tópico k ao atualizar seu vetor de características, como:

$$n_k^{w_{i,1}} = n_k^{w_{i,1}} - 1; n_k^{w_{i,2}} = n_k^{w_{i,2}} - 1; n_k = n_k - 1 \tag{20}$$

Com a técnica de *Sampling Gibbs collapsed*, o BTM obtém amostra do tópico z_i do *biterm* b_i usando o condicional de distribuição descrito na equação (21):

$$p(z_i = k|Z_{\neg i}, B) \propto (n_{k,-i} + \alpha) \times \frac{(n_{k,\neg i}^{w_{i,1}} + \beta)(n_{k,\neg i}^{w_{i,2}} + \beta)}{(n_{k,\neg i} + V\beta + 1)(n_{k,\neg i} + V\beta)} \tag{21}$$

onde $Z_{\neg i}$ denota os tópicos de B, exceto de b_i, e n_k é o número de *biterms* designados para o tópico k.

Primeiro deletam-se cada *biterm* de seu atual vetor de característica por meio das propriedades deletáveis. Então o *biterm* é designado a um novo tópico e o vetor de características é atualizado conforme as propriedades adicionáveis. Após finalizar as iterações, o BTM estima φ e θ, como mostram as equações (22) e (23):

$$\varphi_k^w = \frac{n_k^w + \beta}{n_k + V\beta} \tag{22}$$

$$\theta_d^k = \sum_{i=1}^{n_d^b} p(z_i = k) \tag{23}$$

onde θ_d^k é a probabilidade do tópico k no documento d, e n_d^b é o número de *biterms* no documento d.

Apesar de suas vantagens, o BTM também tem suas limitações. A principal é que a qualidade dos tópicos descobertos pode ser influenciada pela escolha do tamanho da janela de contexto.

3.2.3.2 *Word Network Topic Model* (WNTM)

Zuo *et al.* (2016) apresentam o WNTM, um método que usa a coocorrência global de palavras para construir uma rede de co-ocorrência de palavras, e assim desenvolver a distribuição de cada palavra da rede sobre os tópicos usando o LDA. Além disso, o WNTM considera a dependência entre palavras para melhorar a detecção de tópicos.

A ideia central do WNTM é de que a semântica de uma palavra em um texto curto é fortemente influenciada por suas palavras vizinhas. Portanto, o WNTM constrói uma rede de palavras, em que cada nó representa uma palavra e as arestas representam a dependência semântica entre pares de palavras. Em seguida, o modelo usa essa rede para descobrir tópicos, com a ideia de que palavras conectadas na rede são prováveis de pertencer ao mesmo tópico.

A princípio, define-se uma janela de leitura que se move de palavra a palavra. Dessa forma, duas palavras distintas que apareçam na mesma janela são classificadas como uma coocorrência. O WNTM constrói uma rede não direcionada de coocorrência de palavras, onde cada nó representa uma palavra e o peso de cada aresta é o número de coocorrências das duas palavras conectadas. Portanto, o número de nós da rede equivale ao tamanho do vocabulário V.

Desse modo, o WNTM gera um pseudo documento l para cada vértice v constituído pelos vértices adjacentes de v na rede. A quantidade de ocorrência do vértice adjacente em l é determinado pelo peso da aresta. O número de palavras em l se dá pelo grau do vértice v e o número de pseudo-documentos P é igual ao número de vértices.

Após obter os pseudo documentos P, o WNTM adota o LDA para modelar os tópicos latentes dos pseudo-documentos. Assim, para cada palavra w em l, inferem-se os tópicos usando a distribuição condicional mostrada na equação (24):

$$p(z_{l,w} = k | Z_{\neg(l,w)}, P, \alpha, \beta) \propto (n_{l\neg(l,w)}^k + \alpha) \frac{n_{k,\neg(l,w)}^w + \beta}{n_{k,\neg(l,w)} + V\beta} \tag{24}$$

onde n_l^k são o número de tópicos k que pertencem a l, e $\neg(l,w)$ significa que w foi removido de l.

Dado que cada pseudo documento é a lista de palavras adjacentes de w, então a distribuição tópico-documento aprendida com os pseudo documentos é a base da distribuição de palavras nos tópicos no WNTM. Sendo l gerado a partir de w, a distribuição tópico-palavra de w é calculada conforme a equação (25):

$$\varphi_k^w = \frac{n_l^k + \alpha}{n_l + K\alpha} \tag{25}$$

, onde n_l é o número de palavras em l.

Dada a distribuição tópico-palavra, a distribuição documento-palavra θ_d pode ser calculada como mostra a equação (26):

$$\theta_d^k = \sum_{i=1}^{n_d} \varphi_k^{w_{d,i}} p(w_{d,i}|d) \tag{26}$$

, sendo que

$$p(w_{d,i}|d) = \frac{n_d^{w_{d,i}}}{n_d} \tag{27}$$

, onde $n_d^{w_{d,i}}$ é o número de palavras $w_{d,i}$ no documento d.

Apesar de suas vantagens, o WNTM também tem suas limitações. A construção da rede de palavras pode ser sensível à escolha do método de cálculo da dependência semântica entre as palavras. Além disso, como outros modelos de tópicos, a determinação do número ideal de tópicos a priori pode ser desafiadora. No entanto, o WNTM representa um avanço significativo na modelagem de tópicos considerando a dependência semântica entre as palavras.

3.2.3.3 *Multiterm Topic Model* (MTM)

Uma característica fundamental do BTM é a identificação e formação de *biterms* com pares de palavras. Porém, em determinadas circunstâncias, um par de palavras pode ser insuficiente para representar os padrões formados no texto. Portanto, inspirados no BTM, Xiaobao Wu e Chunping Li (2019) apresentam o MTM, visando apurar se padrões de palavras mais flexíveis podem impactar no desempenho da modelagem de tópicos. Assim, o MTM considera a dependência entre múltiplas palavras simultaneamente para melhorar a detecção de tópicos.

Os autores estabelecem um Multiterm como um conjunto de palavras de tamanho variável e mais relevantes comparados a um *biterm*. Cada *multiterm* abrange um tópico. Ao modelar diretamente os *multiterms* extraídos de cada documento, o MTM pode inferir os componentes do tópico e a distribuição dos tópicos de todo o corpus.

Então, pode-se obter a distribuição de tópicos de todo o corpus, e, com isso, a distribuição de tópicos em cada documento. O modelo MTM se baseia nos padrões de palavras extraídos do corpus, mesmo com os padrões de palavras mais relativos e de tamanhos variáveis.

A extração dos *multiterms* se dá por partes. Primeiro, identificam-se todas as frases nominais, ou seja, toda expressão que não apresenta um verbo na sua composição. Nesse caso, todas as expressões que consistem em adjetivo e substantivo ou apenas substantivo. Os termos que sobram entre as frases nominais também são classificadas como *multiterms m*, representados como $w_1, w_2, ..., w_{|m|}$, onde $|m|$ é o número de palavras distintas no *multiterm m*. O conjunto de *multiterms* em um documento *d* se dá por$M_d = (m_1, m_2, ..., m_{|M_d|})$, em que $\left|M_d\right|$ é o número de *multiterms* em *d*. Após o processamento de todos os documentos, obtém-se o conjunto de *multiterms M* de todo o corpus.

Após a extração de todos os *multiterms* do corpus, começa o processo generativo, assumindo que cada *multiterm* é derivado de um tópico específico. Para tanto, o processo do MTM é demonstrado como:

1. Obter $\theta \sim$ *Dirichlet*(α).
2. Para cada tópico $k \in K$:
    a) Obter $\varphi_z \sim$ *Dirichlet*(β).
3. Para cada multiterm $m \in M$:
    a) Obter $z \sim$ *Multinomial*(θ).
    b) Obter $w_i \sim$ *Multinomial*(φ_z).

, onde α e β são hiperparâmetros da distribuição Dirichlet, θ é a distribuição global de tópicos em todo o corpus, z é a atribuição de tópico de um *multiterm m* e φ_z significa a distribuição de palavras do tópico z. Cada palavra w_i é obtida da distribuição multinomial. A Figura 14 representa graficamente o modelo MTM, onde $\left|M\right|$ refere-se ao número total de todos os *multiterms*.

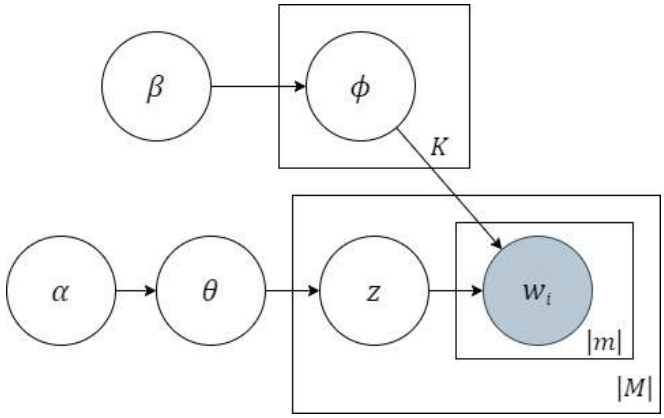
Portanto, com o processo generativo tem-se a probabilidade de um *multiterm m*, representado pela equação (28):

$$p(m) = \sum_z p(z) \prod_{w_i \in m} p(w_i|z) \tag{28}$$

Uma vez que o tamanho do *multiterm* é variável, é preciso considerar a probabilidade de cada palavra nele. Nesse caso, a probabilidade do conjunto de *multiterms M* se dá pela equação (29):

$$p(M) = \prod_m p(m) = \prod_m \sum_z p(z) \prod_{w_i \in m} p(w_i|z) \tag{29}$$

Figura 14 – Representação gráfica do modelo MTM.



Fonte: Adaptado de Xiaobao Wu e Chunping Li (2019).

Com isso, o MTM modela diretamente o processo gerador de *multiterms* em todo o corpus, mesmo eles tendo o comprimento variável. Desta forma, supera-se o problema de esparsidade na modelagem de tópicos de textos curtos.

Uma vez que a distribuição dos tópicos dos *multiterms* já são conhecidas, estes podem ser computados para cada documento. Para inferir $p(z|d)$, assume-se que a distribuição dos tópicos de um documento equivale à expectativa da distribuição de cada *multiterm* gerado no documento. Portanto, a distribuição de tópicos em cada texto curto se dá pela equação (30):

$$p(z|d) = \prod_{m} p(z|m)p(m|d) \tag{30}$$

Com isso, o processo de inferência dos tópicos se dá pelo condicional de distribuição mostrado na equação (31):

$$p(z|z_{-m},M,\alpha,\beta) \propto (n_z + \alpha) \prod_{w_i \in m} \frac{n_{w_i|z} + \beta}{\sum_{w_j} n_{w_j|z} + \left|W\right|\beta} \tag{31}$$

, onde z_{-m} é a atribuição de tópicos de todos os multiterms exceto m, n_z é o número de vezes que m é atribuído a z, $n_{w|z}$ é o número de vezes que a palavra w_i é atribuída a z e $\left|W\right|$ é o tamanho do vocabulário. Com o processo *Gibbs sampling*, obtém-se a distribuição tópico-palavra φ e a distribuição global de tópicos θ, conforme mostram as equações (32) e (33):

$$\varphi_{w_i|z} = \frac{n_{w_i|z} + \beta}{\sum_{w_j} n_{w_j|z} + \left|W\right|\beta} \tag{32}$$

$$\theta_z = \frac{n_z + \alpha}{\left|M\right| + K\alpha} \tag{33}$$

Conforme mostrado no processo generativo, todos os tópicos são atribuídos a cada *multiterm* de M aleatoriamente. Então, obtêm-se os tópicos atribuídos da distribuição condicional $p(z|z_{-m},M,\alpha,\beta)$ e n_z, $n_{w_i|z}$ são atualizados. Por fim, computam-se φ e θ.

Apesar de suas vantagens, o MTM também tem suas limitações. A principal delas é que a escolha do número de palavras em um *multiterm* e a determinação do número ideal de tópicos a priori podem ser desafiantes. Além disso, a complexidade computacional do modelo aumenta com o número de palavras consideradas em cada *multiterm*. No entanto, o MTM ainda representa uma alternativa viável na modelagem de tópicos para textos curtos.

3.2.3.4 *Noise Biterm Topic Model with Word Embeddings* (NBTMWE)

Também inspirado no BTM, Jiajia Huang *et al.* (2020) apresentam o NBTMWE, que obtém os tópicos diretamente de *biterms*. Ademais, considera-se que a frequência e a similaridade semântica dos *biterms* podem ser exploradas para melhorar a qualidade dos tópicos. Este modelo parte do princípio do BTM, e incorpora informações semânticas de palavras por meio de *word embeddings*, além de considerar a presença de ruído nos dados.

O NBTMWE usa *word embeddings* para capturar a semântica das palavras e suas relações contextuais. Ele também incorpora um componente de ruído para modelar palavras que não estão fortemente associadas a nenhum tópico, melhorando a qualidade dos tópicos descobertos. Portanto, ao considerar *biterms* ao invés de documentos individuais, e ao incorporar informações semânticas de palavras e um componente de ruído, o NBTMWE pode descobrir tópicos de alta qualidade em textos curtos.

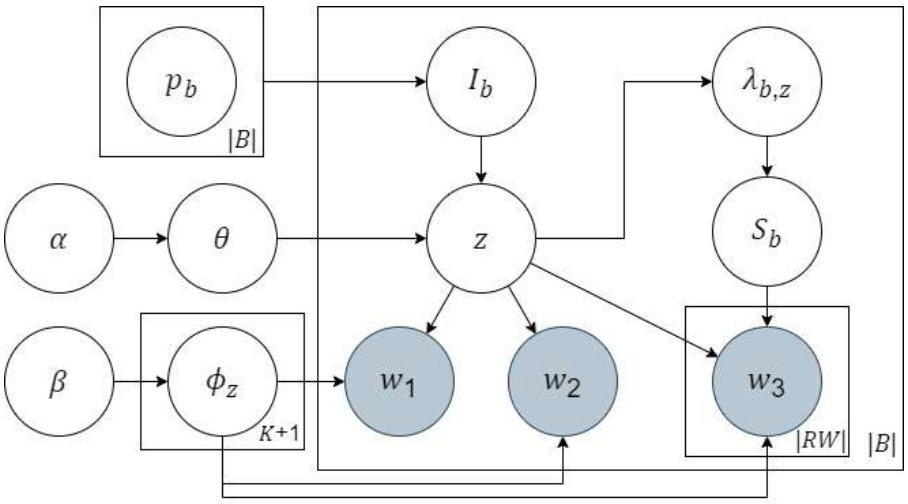
Biterms diferentes possuem capacidades diferentes em agregar palavras similares, onde o *biterm* com maior frequência t_b ou maior similaridade s_b agrega mais palavras similares. Dessa forma, foi proposto uma métrica para medir a probabilidade de ruído de um *biterm*, estimado pelos t_b e s_b do *biterm*, para reunir em um tópico de ruídos os *biterms* com maior probabilidade de ruídos. Quando um *biterm* possui maior valor de probabilidade de ruído p_b, é mais plausível que este tenha se gerado em um tópico de ruídos.

À vista disso, fez-se necessário uma forma de determinar se um *biterm* b é gerado a partir de um tópico de ruídos ou não. Portanto, a distribuição de Bernoulli $I_b \sim Bern(p_b)$ foi adotada. Este é um processo estatístico que se baseia em n tentativas repetidas, no qual a tentativa possui dois resultados possíveis: sucesso ou falha, a probabilidade de sucesso é sempre a mesma em toda tentativa, e a probabilidade de sucesso não é afetada pelo possível conhecimento obtido de resultados anteriores. Nesse caso, se o resultado da distribuição indicar $I_b = 0$, então b foi gerado de um

tópico de ruídos, enquanto $I_b = 1$ indica que b foi gerado de tópicos relevantes.

Por fim, realiza-se uma distribuição multinomial $\varphi_z|z = 0,1,...,K$ sobre o vocabulário para todos os tópicos, onde φ_0 representa a distribuição de palavras sobre o tópico de ruídos e φ_z a distribuição em cada tópico relevante. A Figura 15 representa graficamente o NBTMWE, onde $|B|$ significa o número de biterms em B.

Figura 15 – Representação gráfica do modelo NBTMWE.



Fonte: Adaptado de Jiajia Huang *et al.* (2020).

O processo generativo do conjunto de biterms B no NBTMWE pode ser dividido em duas partes:

(1) Obter distribuição de tópicos relevantes $\theta \sim Dir(\alpha)$

(2) Para cada tópico $z \in [0, K]$:

  (a) Obter distribuição de tópico relevante $\varphi_z \sim Dir(\beta)$

(3) Para cada biterm $b \in B$:

  (a) Obter $I_b \sim Bern(p_b)$

    (i) Se $I_b = 0$:

      (A) Obter duas palavras $w_1, w_2 \sim Multi(\varphi_0)$

    (ii) Senão:

      (A) Obter tópico relevante $z \sim Multi(\theta)$

      (B) Obter promoção de tópico $S_b \sim Bern(\lambda_{b,z})$:

        (1) Se $S_b = 0$:

          (a) Obter duas palavras $w_1, w_2 \sim Multi(\varphi_z)$

        (2) Senão:

(a) Para palavra w_i no conjunto de *biterms* relacionados RW_b:

(i) Obter palavra $w_i \sim Multi(\varphi_z)$

A primeira parte, nos itens 3) e 3).a), determinam se os *biterms* pertencem ao tópico de ruídos pela distribuição de Bernoulli. Após isso, na segunda parte, obtêm-se as palavras para o tópico z. Neste processo, quando um tópico relevante z é designado para b, as palavras semanticamente relacionadas a b são promovidas para o mesmo tópico designado para o *biterm* em questão.

A estratégia de promoção de tópicos é integrada ao processo de amostragem para inferir os tópicos latentes. Assim, é necessário estimar os parâmetros ao substituir uma variável por um valor obtido da distribuição de todas as outras variáveis latentes e dados. Ou seja, a probabilidade de obter um tópico para b depende de duas variáveis latentes, como z_b e l_b, que podem ser estimadas conjuntamente conforme as distribuições condicionais das equações (34) e (35):

$$p(l_b = 0|l^{\neg b}z^{\neg b},B,\alpha,\beta,p_b) = p_b\frac{(n^{\neg b}_{w_l|0})(n^{\neg b}_{w_2|0}+\beta)}{(n^{\neg b}_{0}+V\beta)(n^{\neg b}_{0}+V\beta+1)} \tag{34}$$

$$p(l_b = k,z = k|l^{\neg b}z^{\neg b},B,\alpha,\beta,p_b) = (1-p_b)\frac{(m^{\neg b}_{k}+\alpha)}{\sum_{k=1}^{K}(m^{\neg b}_{k}+\alpha)}\frac{(n^{\neg b}_{w_1|k})(n^{\neg b}_{w_2|k}+\beta)}{(n^{\neg b}_{k}+V\beta)(n^{\neg b}_{k}+V\beta+1)} \tag{35}$$

O processo de amostragem começa com as atribuições de tópicos para cada *biterm*. Em cada iteração do processo, primeiro calcula-se a distribuição condicional de cada tópico, ou seja, os tópicos de ruídos e relevantes. Se b é gerado de um tópico de ruídos, é reduzido o número de palavras em z, caso contrário, diminui-se o número de palavras relacionadas a b com base em S_b. Em seguida, obtêm-se as variáveis latentes de cada *biterm*, segundo as equações (34) e (35). Assim sendo, quando $S_b = 1$, ambas as palavras em b e relacionadas a b vão aumentar a taxa de palavras nos tópicos relevantes. Esse processo continua iterativamente até que o número pré-definido de iterações seja alcançado.

Com isso, este modelo visa melhorar a coerência dos tópicos ao introduzir um tópico de ruídos no modelo e promover ainda mais os padrões de co-ocorrência de palavras e palavras semanticamente semelhantes no mesmo tópico relevante.

Apesar de suas vantagens, o NBTMWE também tem suas limitações. A qualidade das incorporações de palavras e a escolha do método para estimar a proporção de ruído podem afetar a qualidade dos tópicos descobertos. Além disso, como em outros modelos de tópicos, a determinação do número ideal de tópicos a priori pode ser desafiadora. No entanto, o avanço do NBTMWE ainda é relevante na modelagem de tópicos para textos curtos, fornecendo uma abordagem mais robusta e orientada a semântica para a descoberta de tópicos.

3.3 CARACTERÍSTICAS GERAIS

Nugroho *et al.* (2020) fazem considerações significativas sobre os métodos para modelagem de tópicos, onde os métodos tradicionais para modelagem de tópicos funcionam de forma não-supervisionada, considerando apenas o conteúdo presente nos documentos. Muitas das soluções encontradas incluem alternativas que vão desde estender o texto dos documentos, utilizar conhecimentos aprendidos externos ao corpus e até considerar as ocorrências globais de pares de palavras no corpus. Em seu estudo sobre a derivação de tópicos a partir do ambiente do Twitter, Nugroho *et al.* (2020) também destacam que:

- Métodos que dependem inteiramente do conteúdo do documento ainda sofrem com o problema de esparsidade. A densidade de co-ocorrências de palavras em uma coleção de tuítes pode permanecer baixa o suficiente para dificultar a exploração das relações semânticas e derivação de tópicos baseados no conteúdo interno.
- Aumentar o conteúdo dos textos curtos com fontes externas é uma boa solução, mas o conteúdo novo traz consigo novos ruídos e termos sem relação com o contexto dos textos curtos. Lu *et al.* (2017) complementam ao dizer que a efetividade de métodos do tipo serão bastante reduzidas se for utilizado um conjunto de dados de uma fonte sem credibilidade, ou se pouca informação adequada estiver disponível. Além disso, depender de conteúdo externo adiciona uma questão de escalabilidade à solução, trazendo um fardo extra a um problema já existente de lidar com um ambiente tão dinâmico como as mídias sociais.
- Alguns métodos que consideram aspectos sociais se baseiam no uso de elementos como hashtags pelos usuários. Entretanto, isso não resolve o problema de esparsidade, pois a maioria dos tuítes não incluem hashtags. Algumas soluções consideram o autor do tuíte e/ou seus seguidores, porém nos ambientes de mídias sociais os autores não possuem um vínculo forte com os tópicos e um mesmo autor pode publicar diversas vezes sobre vários tópicos.

O uso de *word embeddings* como conhecimento complementar para a modelagem é recurso esporadicamente utilizado. Porém, Fang (2021) aponta alguns problemas em abordagens assim: primeiro, essas tendem a associar palavras semanticamente semelhantes ao mesmo tópico, o que pode ser um problema, pois o tópico resultante pode ser coerente, mas não reflete o que realmente está sendo discutido nos documentos.

O segundo problema é o grau de dificuldade para pessoas não especializadas com a técnica computacional de *embedding* de palavras. Para treinar um modelo

do tipo, é necessário que grandes volumes de amostras de textos em um contexto semelhante sejam coletadas e pré-processadas. Um cientista social, por exemplo, pode deparar-se com uma barreira técnica que pode custar mais tempo que o disponível para ser contornada.

Outro recurso que costuma aparecer nos modelos, são formas de determinar as melhores configurações para os parâmetros do algoritmo. Os métodos de modelagem de tópicos normalmente requerem o número de tópicos como um parâmetro de entrada, e por isso, muitos trabalhos usam aspectos sociais como as hashtags para determinar o número de tópicos. No entanto, outras alternativas devem ser consideradas, pois as hashtags não necessariamente representam efetivamente os tópicos (NUGROHO *et al.*, 2020).

3.4 COMPARAÇÃO ENTRE OS MODELOS

Os algoritmos selecionados para o trabalho foram listados na Tabela 2.

Tabela 2 – Listagem dos algoritmos usados nos experimentos.

Categorias	Algoritmos	Método	Referência
Tradicionais	LDA	Probabilístico, influência bayesiana	Blei, Ng e Jordan (2003)
	LSA	Representação vetorial, Decomposição de Matriz, Análise de valores singulares	Deerwester *et al.* (1990)
	PLSA	Modelo probabilístico, Maximização da verossimilhança	Hofmann (1999)
	NMF	Fatorização de matrizes, Restrição de não-negatividade	Shahnaz *et al.* (2006)
Baseados em DMM	DMM	DMM, *Dirichlet Prior*, *Unigram Model*	Yin e Wang (2014)
	GPU-PDMM	DMM, Modelo *Generalized Polya Urn*, Distribuição de Poisson	Chenliang Li *et al.* (2017)
	NQTM	DMM, *Framework auto-encoding*, Amostragem negativa	Xiaobao Wu *et al.* (2020)
	LapDMM	DMM, *Variational manifold regularization*, Matriz laplaciana	Ximing Li *et al.* (2021)
Auto-agregação	SATM	Auto agregação de documentos, Grafo de documentos	Quan *et al.* (2015)
	CRFTM	*Conditional random field*, Suavização de distribuições	Wang Gao *et al.* (2019)
	PTM	Modelo probabilístico, Agrupamento hierárquico	Zuo *et al.* (2016)
	MIGA	Similaridade de conteúdo, Similaridade de metadados, Aprendizado híbrido	He Zhao *et al.* (2019)
Co-ocorrência global	BTM	Modelo de Bigramas, Co-ocorrência global	Cheng *et al.* (2014)
	WNTM	Rede de palavras (Grafos), *Eigenvector Centrality*	Zuo *et al.* (2016)
	MTM	*Multiterms*, N-gramas	Xiaobao Wu e Chunping Li (2019)
	NBTMWE	Tópico de ruídos, promoção de tópicos	Jiajia Huang *et al.* (2020)

3.5 CONSIDERAÇÕES FINAIS

Este capítulo apresenta os algoritmos de modelagem de tópicos selecionados para a apuração de desempenho proposta neste trabalho. Os trabalhos foram selecionados da literatura com base nas tecnologias adotadas e notoriedade de cada um. A barreira do idioma não foi um problema na escolha, visto que os algoritmos trabalham com métodos estatísticos em suas execuções, sendo muitos deles testados pelos próprios criadores em múltiplos idiomas. Neste trabalho, os testes serão realizados em um corpus com conteúdo exclusivamente classificado como língua portuguesa, para mostrar ainda como os algoritmos performam fora dos idiomas que são majoritariamente utilizados (inglês, chinês) em análises dessa natureza.

4 MÉTRICAS DE AVALIAÇÃO

Abordagens de modelagem de tópicos possibilitam que o conteúdo de tópicos em um corpus possa ser examinado. Para interpretar os tópicos obtidos, é necessário analisar a qualidade interpretativa destes tópicos, e, para isso, uma métrica de qualidade pode avaliar o desempenho de um método de modelagem de tópicos (FANG, 2021).

À vista disso, (RÜDIGER *et al.*, 2022) identificaram desafios na comparação de desempenho entre algoritmos de modelagem de tópicos, especialmente quando se trata da escolha de uma métrica adequada para tal. Os autores expõem outra questão problemática de muitos trabalhos: eles usam data sets muito variados, com diversas características distintas. Por isso, muito desses podem não ser escolhas representativas de dados reais. Portanto, para este estudo, foi utilizado um conjunto de dados reais. Para isso foi preciso definir o escopo do conjunto de dados.

Em uma modelagem, espera-se que os tópicos gerados sejam coesos e de fácil compreensão para humanos, porém nem sempre isso acontece (ALSUMAIT *et al.*, 2009). Um avaliador humano pode medir a qualidade dos tópicos gerados intuitivamente, mas essa pode ser uma tarefa problemática e confusa quando se trata de um corpus muito volumoso, onde se gera um grande número de tópicos (FANG, 2021). Por esse motivo, as qualidades dos algoritmos e de seus resultados devem ser avaliadas.

Conforme descrito anteriormente, cada algoritmo de modelagem de tópicos possui seu próprio design e conjunto de características para realizar o processo de inferência. Assim sendo, para analisar as possibilidades de design de algoritmos é necessário um critério de qualidade, a precisão (RÜDIGER *et al.*, 2022). Os autores também apontam a complexidade computacional e o tempo de cálculo como critérios importantes, porém a precisão mostra-se o mais relevante para avaliar a clusterização que melhor reflete a realidade. A depender das características do algoritmo aplicado e do corpus, uma modelagem pode ser um processo demorado, mas dificilmente ao ponto de ser inviável por esse motivo.

As métricas de avaliação a serem utilizadas dependem da aplicação e das informações disponíveis. Existem dois conceitos de *data mining* que se aplicam para avaliar o resultado de uma modelagem de tópicos (HAN; KAMBER; PEI, 2012; RÜDIGER *et al.*, 2022):

- *Internal cluster validity*, usa medidas que descrevem as propriedades internas de um resultado de clusterização. Considera os aspectos estruturais dos clusters, como seu grau de separação, e não depende de nenhuma informação adicional relacionada aos dados de entrada. Com isso, tem-se uma ideia de qualidade que pode não corresponder à percepção humana. No entanto, em muitos casos, é a única opção viável, pois não existe uma fonte externa

de conhecimento com a qual o agrupamento de textos possa ser comparado (CHANG *et al.*, 2009).

- *External cluster validity*, compara os resultados da clusterização a uma fonte externa de conhecimento, conhecida como *groundthruth*. Normalmente, refere-se a uma rotulagem de clusters feita manualmente. Evidentemente, tal rotulagem manual é baseada em percepções humanas e depende da experiência dos avaliadores. É provável que em um cenário exploratório essa opção esteja indisponível devido à ausência de *groundthruth*.

Para avaliar a qualidade dos tópicos gerados neste trabalho, determinou-se um conjunto de métricas a ser usado. Devido à natureza exploratória dos testes realizados, as métricas são fundamentadas na ideia de *Internal cluster validity*.

4.1 MÉTRICAS INTERNAS DE CLUSTERIZAÇÃO

Dentre as métricas de *Internal cluster validity* disponíveis na literatura, destacam-se: a perplexidade P (WALLACH *et al.*, 2009), a distância de tópicos D_t (CAO *et al.*, 2009), a divergência-KL simétrica D_{KL} (ARUN *et al.*, 2010), e as medidas de coerência C_{UMASS} e C_{W2V} (ROSNER *et al.*, 2014). Essas medidas já foram amplamente usadas em trabalhos acadêmicos e muitas delas são disponibilizadas em pacotes gratuitos de software (ŘEHŮŘEK; SOJKA, 2010; PEDREGOSA *et al.*, 2011; NIKITA, 2016). Neste contexto, coerência refere-se às relações de sentido entre unidades de um texto.

4.1.1 Perplexidade

A avaliação de técnicas de modelagem é uma questão crucial. No entanto, a natureza não supervisionada desses algoritmos torna desafiadora a seleção de um modelo adequado. Em alguns casos, recorre-se ao uso de conhecimento externo como referência para classificar o desempenho. No entanto, essa abordagem nem sempre é viável.

Portanto, Wallach *et al.* (2009) apresentam a perplexidade, uma forma universal, computacionalmente eficiente e independente para medir a capacidade de generalização de uma modelagem de tópicos. A perplexidade é uma métrica comumente usada para avaliar a qualidade de modelos probabilísticos. No contexto da modelagem de tópicos, a perplexidade mede a incerteza do modelo ao prever palavras em um conjunto de dados desconhecido. Em outras palavras, ela indica o quão bem o modelo pode generalizar para novos documentos.

Um modelo de tópicos com perplexidade mais baixa indica uma incerteza menor ao prever palavras em novos documentos. Isso significa que o modelo é capaz de fazer previsões mais precisas e confiáveis, sugerindo que ele capturou os padrões e estruturas subjacentes nos dados de treinamento de maneira eficaz. Em geral, um

modelo com perplexidade mais baixa é considerado melhor ajustado aos dados e, portanto, de melhor qualidade.

Por outro lado, um modelo de tópicos com perplexidade mais alta tem uma incerteza maior ao prever palavras em novos documentos. Isso sugere que o modelo pode não ter aprendido tão bem os padrões e estruturas subjacentes nos dados de treinamento e, portanto, não é tão eficiente na generalização para novos dados. Um modelo com perplexidade mais alta é geralmente considerado de qualidade inferior em comparação com um modelo com perplexidade mais baixa.

Normalmente, o conjunto de dados é dividido em duas partes: conjunto de teste e conjunto de treinamento. Esta é uma métrica preditiva, onde o modelo é treinado com o conjunto de treinamento e então aplicado ao conjunto de teste que contém documentos ainda não vistos. Cada documento pode ser avaliado separadamente, em vista que as atribuições de tópicos para um documento são independentes das atribuições de tópicos para todos os outros documentos. Dessa forma, com evidência empírica os autores demonstraram que o método de perplexidade proposto fornece uma métrica clara para avaliar a performance de uma modelagem de tópicos relativa a outros modelos.

Todavia, ressalta-se que apesar dessa ser uma boa escolha para medir a qualidade, ela não oferece bons resultados no que tange a interpretação humana. Chang *et al.* (2009) mostram que a perplexidade por si não esclarece se os tópicos gerados são coerentes e interpretáveis por humanos, pois esta não captura a relação entre palavras em um tópico ou tópicos de um documento. Os autores observaram que à medida que a pontuação de perplexidade aumenta, a interpretabilidade humana dos tópicos piora.

4.1.2 Distância de tópicos

Na clusterização de tópicos, o número de tópicos é crucial para o desempenho. Entretanto, encontrar um valor apropriado de tópicos não é uma tarefa trivial. Cao *et al.* (2009) propõe um método para identificar a distância de tópicos baseado na densidade. Os autores constataram que ao mapear o processo de geração de um novo tópico, as palavras que conectam vários tópicos provavelmente gerarão os novos tópicos. Além disso, a distância não é apenas determinada pelo tamanho do conjunto de dados, há também a sensibilidade das correlações inerentes à coleção de documentos. Ao computar a densidade de cada tópico, encontra-se os mais instáveis, e iterativamente atualiza-se a quantidade de tópicos até atingir um modelo estável. Em suma, a distância de tópicos refere-se à medida de semelhança entre dois tópicos em um espaço multidimensional.

Numa avaliação de distância entre os tópicos, quando os valores retornados são baixos, significa que os tópicos são semanticamente semelhantes ou próximos

uns dos outros. Essa proximidade pode indicar que os tópicos têm muitas palavras ou conceitos em comum, sugerindo que os tópicos podem ser redundantes ou que o modelo não conseguiu separar adequadamente os tópicos distintos presentes nos dados.

Por outro lado, quando a distância de tópicos é alta, isso implica que os tópicos são semanticamente distintos e bem separados. Isso é desejável em um modelo de tópicos por indicar que os tópicos aprendidos capturam diferentes aspectos ou temas presentes nos dados e não há sobreposição excessiva entre os tópicos. Uma maior distância de tópicos geralmente sugere que o modelo pode identificar e representar adequadamente a estrutura latente nos dados.

No entanto, a preferência por distância de tópicos alta ou baixa em modelagem de tópicos pode depender dos objetivos específicos e das características do conjunto de dados. No caso de uma análise exploratória dos textos, onde o objetivo é explorar e identificar temas distintos em um conjunto de dados, tópicos bem separados podem fornecer ideias mais claras e úteis. Já no caso de uma análise de subtemas, onde se analisa as nuances em um tema mais amplo, a distância de tópicos mais baixa pode ajudar a identificar esses subtemas.

O método é baseado em estatísticas de todo o corpus, sem extensões para amostras de fontes externas. A similaridade por cosseno é usada como indicador de coesão entre textos, para medir a correlação entre tópicos. Em seguida, adota-se no método o conceito apresentado por Ester *et al.* (1996), de clusterização baseada na densidade para selecionar adaptativamente o número de tópicos, com base na densidade dos tópicos. A ideia de clusterização baseado na densidade é que a similaridade será a maior possível no intra-cluster, mas a menor possível entre clusters distintos.

Assim, identifica-se a melhor estrutura de tópicos, dado que um tópico é um cluster semântico. Uma maior similaridade no intra-cluster mostra que esse cluster representa um significado mais interpretável, e uma similaridade menor entre clusters significa que a estrutura de tópicos é mais estável. Os experimentos mostram que o método é efetivo, e performa melhor quando a similaridade por cosseno entre tópicos atinge o mínimo.

4.1.3 Divergência-KL simétrica

Baseado na divergência de Kullback-Leibler, Arun *et al.* (2010) apresentam uma medida de dissimilaridade entre duas distribuições de probabilidade. O algoritmo de modelagem de tópicos é interpretado como um mecanismo de fatorização de matriz, onde o corpus é dividido em dois fatores matriciais.

A métrica de divergência-KL simétrica baseia-se na divergência de Kullback-Leibler, uma medida não simétrica que quantifica a diferença entre duas distribuições ao calcular o custo adicional de informações necessárias para aproximar uma distri-

buição pela outra. Então, a divergência-KL simétrica é obtida ao calcular a média das divergências de Kullback-Leibler em ambas as direções (de P para Q e de Q para P, sendo eles dois documentos distintos).

A divergência-KL simétrica é uma medida que compara duas distribuições de probabilidade e é frequentemente utilizada no contexto de modelagem de tópicos para avaliar a dissimilaridade entre as distribuições de probabilidade dos tópicos.

Quando falamos de divergência-KL simétrica mais alta no contexto de modelagem de tópicos, estamos nos referindo a uma maior dissimilaridade entre as distribuições de probabilidade dos tópicos. Em outras palavras, os tópicos são distintos e abordam temas diferentes, sem muita sobreposição de termos. Isso pode ser útil quando se deseja garantir que os tópicos identificados sejam claramente separados e representem diferentes áreas de interesse.

Por outro lado, a divergência-KL simétrica mais baixa indica uma menor dissimilaridade entre as distribuições de probabilidade dos tópicos. Isso sugere que os tópicos compartilham termos e padrões semelhantes, o que pode ser interpretado como uma maior similaridade entre os temas abordados. Uma divergência-KL simétrica baixa pode ser desejável em situações em que se espera encontrar tópicos com características semelhantes, como ao analisar documentos relacionados a um tema específico.

Essa métrica é mais intuitiva e fornece uma medida mais balanceada de dissimilaridade por considerar as diferenças entre as distribuições de maneira simétrica, garantindo que a divergência-KL simétrica entre P e Q seja igual à divergência-KL simétrica entre Q e P. Essa propriedade é especialmente útil ao comparar e avaliar modelos e distribuições em diversos contextos, como na modelagem de tópicos e aprendizado de máquina.

Para isso, um método de fatorização matricial divide uma matriz de frequência de documentos M em duas matrizes $M1$ e $M2$ de ordem $T \times W$ e $D \times T$ respectivamente, onde T é o número de tópicos, W é o tamanho do vocabulário do corpus e D o conjunto de documentos. Assim, a calcula-se a divergência simétrica de Kullback-Leibler das distribuições de valores singulares da matriz $M1$ e a distribuição do vetor $L \times M2$ onde L é um vetor $1 \times D$ contendo os comprimentos de cada documento no corpus.

A eficiência da métrica é mostrada aplicando-a em conjuntos de dados reais e sintéticos. Os testes mostram que as distribuições são comparáveis e ainda podem estimar um número adequado de tópicos. Nesse caso, o número de tópicos considerado ideal é qualquer número em um pequeno intervalo que dá a melhor precisão em um conjunto de dados não visto.

4.1.4 Medidas de coerência

Um texto coerente permite o reconhecimento imediato de suas relações semânticas. Medidas de coerência de tópicos baseadas em palavras também são comumente usadas na avaliação de modelagens. Visto que os algoritmos de modelagem descrevem os tópicos encontrados como listas de palavras, a coerência determina a interpretabilidade semântica dos tópicos descobertos, isto é, como as palavras estão conectadas entre si (LAU; NEWMAN; BALDWIN, 2014).

O trabalho de Chang *et al.* (2009) analisou um problema de intrusão de palavras com avaliadores humanos. No teste, os avaliadores deveriam encontrar uma palavra não relacionada em meio a um conjunto de palavras caracterizando um tópico. Por fim, constatou-se que a noção humana de tópicos coerentes não correlaciona bem com a perplexidade. À vista disso, os autores indicam a necessidade de empregar medidas semânticas na análise de qualidade do modelo de tópico.

4.1.4.1 C_{UMASS}

Mimno *et al.* (2011) apresentam uma métrica de avaliação automatizada para identificar tópicos sem depender de conhecimento além dos dados de treinamento. Para isso, aplica uma probabilidade logarítmica condicional de palavras, estimada com base nas frequências dos documentos do corpus original (C_{UMASS}).

Essa métrica de coerência baseia-se apenas em estatísticas de co-ocorrência de palavras coletadas do corpus que está sendo modelado e não depende de um corpus externo de referência. O desempenho da métrica C_{UMASS} implica que não importa a diferença entre a probabilidade conjunta de um par de palavras, mas sim a probabilidade condicional de cada uma das palavras de classificação mais alta no tópico.

A coerência C_{UMASS} é uma métrica que avalia a qualidade dos tópicos gerados ao medir o quão coerentes são as palavras em um tópico. Ela faz isso calculando a média do logaritmo das razões entre a coocorrência de pares de palavras do tópico e a frequência de cada palavra individualmente. Essa métrica se baseia na premissa de que palavras que ocorrem juntas com frequência no corpus devem compor tópicos coerentes.

No contexto da modelagem de tópicos, resultados mais altos para a métrica C_{UMASS} indicam maior coerência entre as palavras do tópico, o que sugere que o algoritmo identificou com sucesso relações semânticas e estruturais relevantes entre as palavras. Por outro lado, resultados mais baixos para essa métrica indicam menor coerência entre as palavras do tópico, o que pode sugerir que o algoritmo não capturou relações significativas e, portanto, pode ter gerado tópicos menos interpretáveis e úteis.

Em geral, é desejável obter alta coerência C_{UMASS} na modelagem de tópicos,

pois isso indica tópicos com palavras semanticamente relacionadas e, portanto, mais interpretáveis e úteis. A coerência alta é particularmente importante é quando os tópicos gerados são usados para análises e tomada de decisões, como em análise de sentimentos, recomendação de conteúdo ou classificação de documentos. Nesses casos, tópicos coerentes facilitam a compreensão e interpretação dos resultados pelos usuários ou por outros sistemas.

No entanto, ressalta-se que, em algumas situações, uma coerência C_{UMASS} baixa pode ser aceitável ou até mesmo desejável. Por exemplo, ao explorar novas áreas de pesquisa ou analisar conjuntos de dados com relações semânticas não óbvias entre palavras, uma baixa coerência pode revelar tópicos inesperados e interessantes que podem levar a novas visões e abordagens. Além disso, em cenários onde a geração de tópicos é mais focada em encontrar padrões de distribuição de palavras em vez de identificar tópicos semanticamente coerentes, uma coerência mais baixa pode ser aceitável.

Para avaliar a capacidade da coerência C_{UMASS}, Mimno *et al.* (2011) replicaram a avaliação de "intrusão de palavras" originalmente introduzida por Chang *et al.* (2009). Dez avaliadores humanos deveriam identificar palavras intrusas em um conjunto de tópicos previamente examinados. Nesta tarefa, uma das dez palavras de um tópico é substituída por outra palavra, selecionada aleatoriamente do corpus. Em tópicos classificados como coerentes os avaliadores conseguiram detectar a palavra intrusa com alta precisão. Em tópicos ruins a precisão de detecção de intrusos foi baixa, com frequências semelhantes. Os resultados sugerem ser mais difícil detectar intrusos em tópicos com baixa coerência.

Com isso, determinou-se o seguinte: (1) há uma classe de tópicos de baixa qualidade que não podem ser detectados usando testes de intrusão de palavras existentes, mas que podem ser identificados de forma confiável usando uma métrica baseada em estatísticas de co-ocorrência de palavras; (2) é possível elevar a pontuação geral de coerência dos tópicos, mesmo para os dez piores, ao mesmo tempo em que se preserva a capacidade de detectar os tópicos ruins, sem depender de dados semi-supervisionados ou fontes externas de referências. Embora informações adicionais possam ser úteis, elas não são necessárias.

4.1.4.2 C_{W2V}

O trabalho de O'Callaghan *et al.* (2015), analisa e compara tópicos encontrados por variantes populares de algoritmos tradicionais em vários corpus. A coerência foi aferida usando uma combinação de medidas existentes e novas, incluindo uma baseada em semântica distributiva. Para isso, os autores compilaram seis corpus, novos e já existentes, contendo documentos que haviam sido manualmente anotados em classes, incluindo artigos de notícias online da BBC, do Guardian e do New York Times, além

do conteúdo de artigos da Wikipedia. Para calcular as frequências de co-ocorrência, usou-se o corpus sendo modelado, em vez de se basear em um corpus de referência.

A coerência C_{W2V} é uma métrica que avalia a qualidade dos tópicos gerados por modelos de tópicos, considerando a similaridade semântica entre as palavras mais representativas de cada tópico. Ela utiliza vetores *Word2Vec*, representações vetoriais de palavras que capturam informações semânticas e contextuais, para calcular a similaridade entre as palavras. Para calcular a coerência C_{W2V}, utiliza-se um modelo *Word2Vec* (MIKOLOV *et al.*, 2013), onde vetores de palavras mais representativas de um tópico são obtidos. Em seguida, a similaridade de cosseno entre os pares desses vetores é calculada com um modelo Skip-gram, e a média dessas similaridades é tomada como a medida de coerência C_{W2V} para o tópico em questão.

No contexto da modelagem de tópicos, resultados mais altos de coerência C_{W2V} indicam tópicos mais coerentes e significativos, pois as palavras representativas estão mais fortemente relacionadas semanticamente. Isso facilita a interpretação e a identificação do tema subjacente do tópico. Já resultados mais baixos de coerência C_{W2V} apontam para tópicos menos coerentes e possivelmente mais difíceis de entender, pois as palavras representativas não compartilham relações semânticas claras ou fortes.

Na modelagem de tópicos, geralmente, a coerência C_{W2V} alta é mais desejável em quase todos os contextos, por indicar que os tópicos gerados são mais coerentes e semanticamente relacionados, tornando-os mais interpretáveis e úteis. Tópicos coerentes podem melhorar a análise de documentos, agrupamento, recuperação de informações e outras aplicações que envolvam a organização e a compreensão de grandes conjuntos de textos.

No entanto, pode haver contextos específicos em que C_{W2V} mais baixa seja desejável. Por exemplo, em situações onde os documentos são intencionalmente compostos por palavras não relacionadas ou têm misturas de tópicos muito distintos, a coerência mais baixa pode ser um sinal de que o modelo de tópicos está capturando corretamente essas diferenças. Além disso, em cenários exploratórios, onde o objetivo é identificar tópicos inesperados ou não convencionais, uma C_{W2V} mais baixa pode sugerir a presença de tópicos incomuns que merecem uma investigação mais aprofundada.

Essa ferramenta fornece dois algoritmos baseados em rede neural para estimar representações de palavras em um espaço vetorial: *continuous bag-of-words*, onde a palavra atual é prevista com base em seu contexto; e Skip-gram, que prevê palavras de contexto com base na palavra atual. Descobriu-se que essas abordagens geram vetores de palavras que codificam explicitamente regularidades linguísticas de grandes quantidades de dados de texto não estruturados (MIKOLOV *et al.*, 2013). Portanto, são apropriados para uso com um grande corpus de referência na análise da coerência do

tópico.

Observou-se que um papel fundamental é desempenhado pela estratégia de ponderação de termos associada, onde modificações no pré-processamento do termo do documento e no pós-processamento do termo descritor podem produzir resultados nitidamente diferentes. Por fim, os autores constataram que independente da técnica de modelagem de tópicos empregada, é essencial uma leitura atenta de quaisquer tópicos gerados.

4.2 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as métricas de avaliação selecionadas para avaliar o desempenho dos algoritmos de modelagem de tópicos analisados neste trabalho. As métricas foram escolhidas da literatura com base em suas capacidades de avaliar determinados aspectos de cada modelo, além de terem sido amplamente empregadas em outras pesquisas como forma de aferição de desempenho para algoritmos de modelagem.

Para este trabalho, mostrou-se necessário priorizar as métricas de avaliação que não usam conhecimento externo. Visto que este trabalho envolve textos de mídias sociais em português sobre COVID-19, a deficiência de bases de dados dessa categoria e idioma impediram que uma fonte de conhecimento externo com qualidade comprovada pudesse ser utilizada para as avaliações propostas nessa pesquisa.

5 PROCEDIMENTOS DE ANÁLISE DOS ALGORITMOS

5.1 INTRODUÇÃO

As plataformas de mídia social desempenham um papel fundamental como palcos para debates acerca de eventos de grande repercussão. O Twitter, em particular, é projetado para incentivar os usuários a discutir e expressar suas opiniões sobre temas em destaque (SANTAELLA; LEMOS, 2010). De acordo com as autoras, a singular dinâmica do Twitter possibilita a rápida e viral disseminação de ideias, refletindo os pensamentos e ideias de um amplo público engajado nas discussões. Nesse contexto, o conteúdo gerado nessas plataformas pode representar a voz coletiva dos usuários, enfatizando a importância da análise dessas interações para a compreensão das dinâmicas comunicacionais contemporâneas.

Ao longo dos anos, o Twitter tem se mostrado uma fonte crucial para mineradores de dados e linguistas interessados em obter dados reais de mídias sociais. A estrutura de microblogging da plataforma possibilita que qualquer evento possa ser compartilhado e comentado entre seus usuários em pouco tempo após seu acontecimento. Posteriormente, todas essas discussões ficam disponíveis publicamente na plataforma, desde que as mensagens não sejam excluídas em algum momento.

O conteúdo dos textos publicados pelos usuários tem o potencial de revelar suas reações aos acontecimentos e suas posições em relação aos eventos. Isso abre novas oportunidades para identificar os temas que permeiam as discussões realizadas online. Nesse sentido, a modelagem de tópicos surge como um método amplamente utilizado para analisar grandes volumes de texto, extraindo informações ocultas. Seu propósito é identificar conjuntos de palavras semanticamente relevantes presentes no corpus de textos, ou seja, os tópicos em destaque.

Considerando a maneira rápida e concisa com que as pessoas compartilham suas ideias nas mídias sociais, é importante considerar a abordagem adequada para inferir tópicos nesse contexto. Vários estudos na literatura têm proposto soluções para lidar com desafios como a esparsidade dos dados, que é inerente a esse tipo de conteúdo. No entanto, muitas pesquisas ainda utilizam algoritmos tradicionais, como o LDA, para a modelagem de tópicos em textos curtos.

Nesse sentido, o objetivo deste estudo foi analisar diversos algoritmos de modelagem de tópicos, comparando a qualidade e coerência dos tópicos gerados em três cenários de conjuntos de dados com características distintas. O foco foi investigar o desempenho de diferentes técnicas de modelagem ao analisar textos originados de mídias sociais. Dessa forma, busca-se contribuir para o desenvolvimento de soluções mais eficientes na inferência de tópicos em textos curtos presentes nessas plataformas.

Desta forma, foi estabelecido um esquema para analisar os tópicos de interesse nas discussões online sobre COVID-19, especificamente provenientes da plataforma

do Twitter. Para realizar a modelagem de tópicos nos textos dessa plataforma de mídia social, o processo foi dividido em quatro etapas: (1) coleta dos dados da plataforma; (2) seleção de amostras dos dados coletados a serem utilizados; (3) aplicação de uma série de tratamentos nos documentos selecionados e; (4) aplicação do algoritmo para modelagem dos tópicos. Estas etapas são descritas nas seções seguintes.

5.2 COLETA E SELEÇÃO DE TEXTOS

Para este estudo, utilizou-se um conjunto de dados reais visando comparar os algoritmos de modelagem de tópicos. Foi utilizado um conjunto de publicações em português coletados do Twitter, os quais foram formados a partir da seleção de termos, hashtags e palavras-chave associadas a discussões relacionadas à COVID-19.

A criação desse conjunto deu-se a partir do trabalho de Banda *et al.* (2021), que coletou e reuniu tuítes publicados pelos usuários do Twitter em tempo real. Desde janeiro de 2020, foram coletados mais de um bilhão de tuítes relacionados à COVID-19 em mais de 60 idiomas, inclusive em português. Os autores coletaram tuítes com base em termos e hashtags associadas à COVID-19, como #coronavirus, #corona, #COVID-19, entre muitos outros. Com a evolução da pandemia novos termos surgiram e foram incorporados ao processo de pesquisa do trabalho. Todos os tuítes elegíveis foram coletados e processados para um formato que pudessem ser compartilhados, conforme as políticas de privacidade do Twitter. Outros pesquisadores contribuíram pontualmente na construção do conjunto de dados, especialmente em idiomas não-inglês e com dados do início da pandemia, quando ainda não havia uma coleta dedicada de tuítes.

A escolha de utilizar o Twitter como fonte de dados baseou-se na vasta coleção de textos curtos gerados pelos usuários, que permite a extração de uma quantidade significativa de textos reais. Além disso, o uso de hashtags e outros recursos de indexação da plataforma possibilitaram a coleta de textos relacionados a temas específicos.

Após a coleta dos dados, foram selecionados apenas os tuítes em português no período de 1 de março de 2020 até 1 de março de 2021. Considerando que retuítes são republicações de outros tuítes, eles não foram considerados para essa pesquisa.

5.3 PRÉ-PROCESSAMENTO DOS DADOS

Um passo crítico na análise textual é o pré-processamento do texto. Esse procedimento envolve uma série de ações para limpar e normalizar o texto para remover possíveis ruídos e maximizar a qualidade dos resultados. Normalmente, os textos são carregados de elementos desnecessários para uma análise automatizada e isso se dá especialmente em mídias sociais que possuem elementos textuais que não se aproveitam para a modelagem de tópicos, como menções a usuários, ou URLs.

O pipeline de pré-processamento começou convertendo todos os documentos para letras minúsculas, exceto para termos escritos completamente em letras maiúsculas com comprimento de dois ou mais caracteres. Para essa filtragem consideram-se elementos como: símbolos de hashtag; menções a usuários; URLS; caracteres que não fazem parte da tabela ASCII; caracteres especiais; pontuação; stop words (não contribuem no significado semântico das mensagens); tags HTML; emojis e outros símbolos não-alfanuméricos; números isolados, palavras repetidas no mesmo documento.

Além disso, uma das características gramaticais da língua portuguesa e outros idiomas, são as diversas formas e variações que cada palavra pode possuir. Termos como “infectados” e “infectada” representam uma única ideia, contudo, na prática, são consideradas elementos distintos no momento da modelagem de tópicos. Para contornar esse problema, a lematização analisa cada palavra individualmente para reduzi-la ao seu radical, retirando as inflexões e resultando em uma palavra válida na gramática (DE LUCCA; NUNES, 2002). Assim, ambos “infectada” e “infectados” resultam em “infectado”. Dessa forma, todos os termos do corpus foram reduzidos aos seus respectivos lemas, excluindo as palavras totalmente maiúsculas.

Palavras que aparecem com pouca frequência no conjunto de textos são consideradas irrelevantes, portanto, são filtradas. Enquanto as palavras que aparecem com alta frequência, acima de 60% dos documentos, também precisam ser removidas, pois estas interferem diretamente no resultado da modelagem. Visto que aparecem muito em vários documentos, elas são interpretadas como semanticamente relevantes em vários tópicos e no fim são “beneficiadas” em diversos deles.

É possível que o resultado do pré-processamento faça o conteúdo de alguns documentos ficarem iguais. Nesse caso, não há necessidade de analisar mensagens repetidas, especialmente as advindas de mídias sociais onde usuários e agentes automatizados disseminam SPAM que potencialmente interfere no resultado da análise de texto (LEDFORD, 2020). Para criar condições comparáveis para todos os métodos testados, os mesmos textos igualmente pré-processados foram usados por todos os algoritmos testados.

5.4 MATERIAIS

Este capítulo relata os materiais utilizados para essa pesquisa. Os 16 algoritmos foram implementados usando a linguagem de programação Python 3.9.7, sendo que para alguns casos as bibliotecas Gensim (LDA, LSA), scikit-learn (NMF) e Bitermplus (BTM) já forneciam implementações para algum método. Além disso, foi necessário implementar estratégias de otimização como paralelização e matrizes esparsas para viabilizar o funcionamento dos algoritmos.

Essas ações foram necessárias, pois os algoritmos de modelagem de tópicos possuem, em geral, um custo computacional elevado. Para mensurar a complexidade

de tempo para cada método, considera-se o número de iterações (10 em todos os casos), o número de documentos (3.941.033), número de tópicos (variou de 10 a 100), e número de palavras (37.463.729). Nesse caso, a complexidade de tempo total é O(n_iter $\times$ n_docs $\times$ n_topicos $\times$ n_palavras). Além disso, alguns algoritmos ainda possuem características próprias que adicionam ainda mais para a complexidade.

Para executar os algoritmos, a máquina utilizada possui um processador Intel Core i7-10750H com 12 núcleos e 16 GB de RAM.

6 RESULTADOS E DISCUSSÃO

Neste capítulo, é apresentada uma discussão a respeito dos resultados obtidos na apuração dos algoritmos de modelagem de tópicos analisados neste trabalho. A discussão abrange uma visão geral sobre os resultados obtidos para cada métrica, com análises de tendências gerais e estabilidade dos algoritmos observadas nos resultados, e comparação entre as categorias de algoritmos. Os resultados individuais de cada avaliação estão detalhadamente expostos no Apêndice A.

O objetivo desse capítulo é entender o desempenho geral dos algoritmos em um contexto mais amplo ao identificar as diferenças dos algoritmos em termos de performance, e como as mudanças de parâmetro os afetaram. Em seguida, é feita uma análise sobre relações notáveis entre a distância de tópicos e divergência-KL simétrica em alguns casos. E por fim, uma seção com considerações gerais sobre os resultados obtidos no estudo.

6.1 PERPLEXIDADE

A métrica de perplexidade é uma medida quantitativa usada para avaliar a qualidade de modelos probabilísticos de linguagem, como os algoritmos de modelagem de tópicos. O princípio fundamental da perplexidade é medir o quão bem o modelo consegue prever um conjunto de dados de teste, comparando as probabilidades atribuídas pelo modelo às palavras reais nesse conjunto.

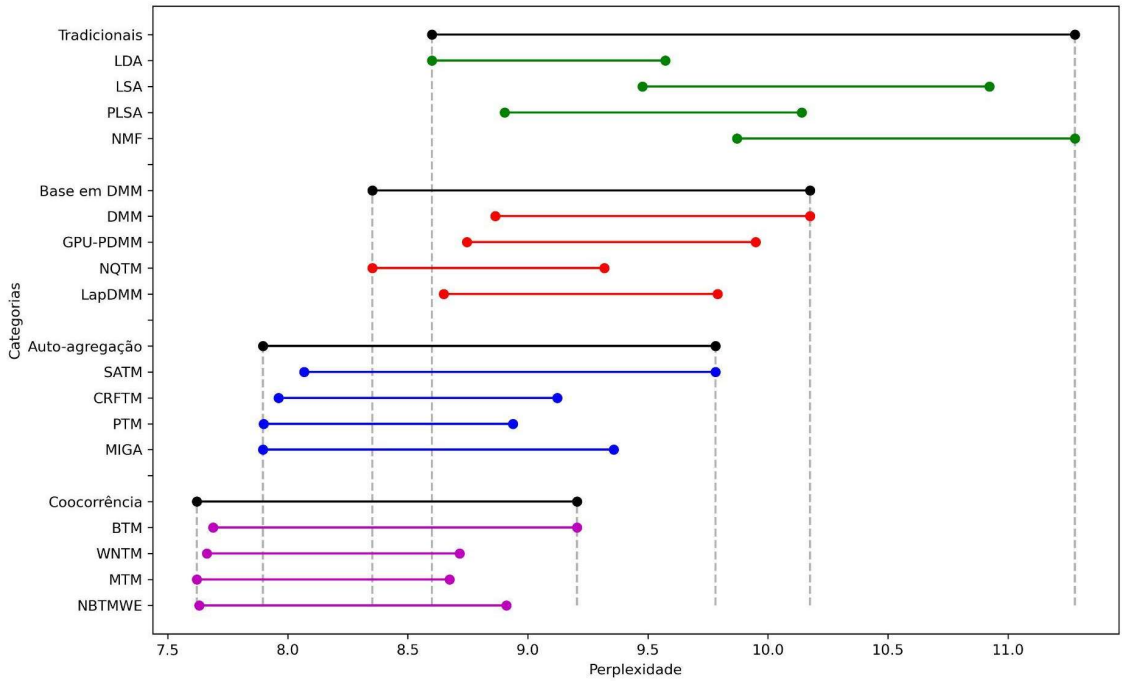
Em outras palavras, a perplexidade mede a incerteza do modelo ao fazer previsões, e um valor menor de perplexidade indica um modelo mais bem ajustado e eficiente na previsão das palavras em um conjunto de dados desconhecido.

Tendência geral: em todas as categorias a perplexidade tende a aumentar à medida que o número de tópicos aumenta. Isso mostra que a qualidade do ajuste do modelo diminui à medida que mais tópicos são adicionados, pois mais tópicos podem levar a uma maior complexidade no ajuste do modelo. No entanto, ressalta-se que a taxa de aumento da perplexidade varia entre os algoritmos e categorias.

Estabilidade dos algoritmos: todos os algoritmos apresentam um crescimento estável em termos de perplexidade, à medida que o número de tópicos aumenta. Alguns casos se destacam, como o NQTM (baseado em DMM) e o PTM (baseado em autoagregação), que mostram um aumento mais lento na perplexidade em comparação com outros algoritmos em suas respectivas categorias. Isso sugere que esses algoritmos podem ser mais robustos em relação às mudanças no número de tópicos. Dado que as avaliações foram realizadas 10 vezes em cada algoritmo, os resultados de cada uma foi representada como intervalos na Figura 16.

Para representar as variações de resultados de cada algoritmo, as linhas na Figura 16 representam o menor valor observado em suas extremidades esquerdas, e o

Figura 16 – Intervalos de perplexidade.



maior valor na extremidade direita. Para cada categoria de algoritmo, seus respectivos intervalos estão representados com as linhas pretas, e o conjunto de algoritmos da mesma categoria possui uma cor única.

Os algoritmos Tradicionais variaram de 8,6 a 11,3; os baseados em DMM tiveram variação de 8,4 a 10,2; os de autoagregação apresentaram 7,9 a 9,8; enquanto os de coocorrência global variaram de 7,6 a 9,2.

Comparação entre categorias: os algoritmos baseados em autoagregação e coocorrência global tendem a ter valores de perplexidade menores em comparação aos algoritmos tradicionais e baseados em DMM. Isso indica que os modelos baseados em autoagregação e coocorrência global podem se ajustar melhor aos textos curtos em comparação com os outros dois tipos de modelos.

Algoritmos com menor perplexidade: dentro de cada categoria, foram identificados os algoritmos com menor perplexidade:

- Tradicionais: LDA
- Baseados em DMM: NQTM
- Baseados em autoagregação: MIGA, PTM
- Baseados em coocorrência global: MTM, NBTMWE

Algoritmos com maior perplexidade: há também os algoritmos com maior perplexidade:

- Tradicionais: NMF
- Baseados em DMM: DMM
- Baseados em autoagregação: SATM
- Baseados em coocorrência global: BTM

Em resumo, observou-se que os algoritmos baseados em autoagregação e coocorrência global tendem a generalizar melhor seus modelos em comparação aos algoritmos tradicionais e baseados em DMM. Apesar disso, um comportamento notável entre todos os algoritmos foi o aumento da perplexidade conforme aumentava o número de tópicos gerados. Esse comportamento condiz com o que foi observado por Hagen *et al.* (2015), que constatou um desempenho semelhante na distribuição de palavras nos tópicos, que se tornava menos focada e menos distinta conforme aumentou a complexidade do modelo.

6.2 DISTÂNCIA DE TÓPICOS

A métrica de distância de tópicos é uma medida quantitativa que avalia a similaridade entre dois tópicos em um espaço multidimensional. Essa métrica pode ser utilizada para analisar e comparar a qualidade de modelos de tópicos, identificando a separação média entre os tópicos aprendidos. A distância de tópicos auxilia na interpretação dos resultados de modelagem de tópicos, fornecendo uma compreensão mais clara da estrutura latente descoberta nos dados e ajudando a identificar redundâncias ou sobreposições entre os tópicos aprendidos.

Uma distância menor entre os tópicos indica que eles são semanticamente semelhantes, enquanto uma distância maior sugere que os tópicos são distintos e bem separados.

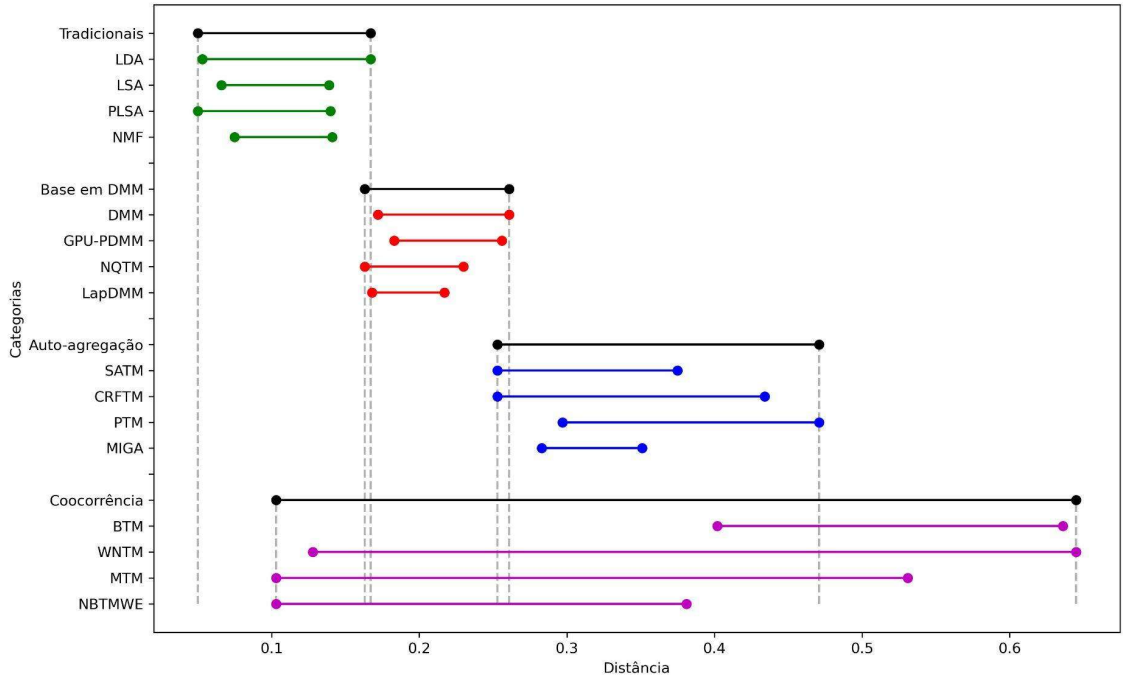
Tendências gerais: os resultados mostram que os algoritmos de autoagregação e coocorrência global são mais propensos a produzir tópicos distintos. Em geral, observou-se que os esses algoritmos apresentam valores mais altos de distância de tópicos, enquanto os algoritmos tradicionais e baseados em DMM tenderam a valores mais baixos.

Estabilidade dos algoritmos: a Figura 17 mostra os intervalos de Distância de cada algoritmo.

Os algoritmos tradicionais (exceto LDA) e baseados em DMM se mostraram mais estáveis na geração de tópicos ao apresentarem variações menores nos resultados, com variações de 0,05 a 0,16 e 0,16 a 0,26 respectivamente. Por outro lado, os algoritmos LDA, baseados em autoagregação (variação de 0,25 a 0,47) e coocorrência global (variação de 0,10 a 0,64) apresentaram uma variação maior nas distâncias.

Comparação entre categorias: os algoritmos Tradicionais apresentaram valores de distância de tópicos relativamente baixos. Os baseados em DMM mostraram

Figura 17 – Intervalos de distância de tópicos.



um aumento na distância de tópicos em comparação com os Tradicionais. Os algoritmos baseados em coocorrência apresentaram distâncias de tópicos ainda mais altas, especialmente o BTM. Os outros algoritmos obtiveram uma variação maior entre os resultados. Por fim, os algoritmos baseados em autoagregação apresentaram, em geral, as maiores distâncias de tópicos entre todas as categorias. Apesar disso, o BTM (coocorrência) mostrou o maior desempenho individual, mantendo a consistência em todas as quantidades de tópicos.

Algoritmos com maior distinção de tópicos: dentro de cada categoria, em geral, os algoritmos que apresentaram maior distância entre tópicos, (e, portanto, tópicos mais distintos) são:

- Tradicionais: NMF
- Baseados em DMM: DMM, GPU-PDMM
- Baseados em autoagregação: PTM
- Baseados em coocorrência global: BTM, WNTM

Algoritmos com menor distinção de tópicos: em geral, os algoritmos que apresentam menor distância entre tópicos (e, portanto, tópicos mais semelhantes) são:

- Tradicionais: PLSA

- Baseados em DMM: NQTM, LapDMM
- Baseados em autoagregação: SATM, CRFTM
- Baseados em coocorrência global: MTM, NBTMWE

Em resumo, os algoritmos Tradicionais e baseados em DMM se mostraram mais estáveis quanto a distância de tópicos. Os Tradicionais, em especial, usam abordagens clássicas que compartilham entre si muitas características semelhantes na geração dos tópicos (ALGHAMDI; ALFALQI, 2015; KHERWA; BANSAL, 2019). Essas semelhanças estruturais dos algoritmos se refletiram nos resultados, onde em vários casos esses apresentaram performances semelhantes. Ainda assim, apesar da maior sensibilidade aos ajustes do modelo e dos parâmetros, os baseados em autoagregação e coocorrência global mostraram desempenho maior, isto é, maior distância entre os tópicos gerados por eles.

6.3 DIVERGÊNCIA-KL SIMÉTRICA

A métrica de divergência-KL simétrica é uma medida de dissimilaridade entre duas distribuições de probabilidade. Baseia-se na divergência de Kullback-Leibler, uma medida não simétrica que quantifica a diferença entre duas distribuições ao calcular o custo adicional de informações necessárias para aproximar uma distribuição pela outra. Essa propriedade é especialmente útil ao comparar e avaliar modelos e distribuições em diversos contextos, como na modelagem de tópicos.

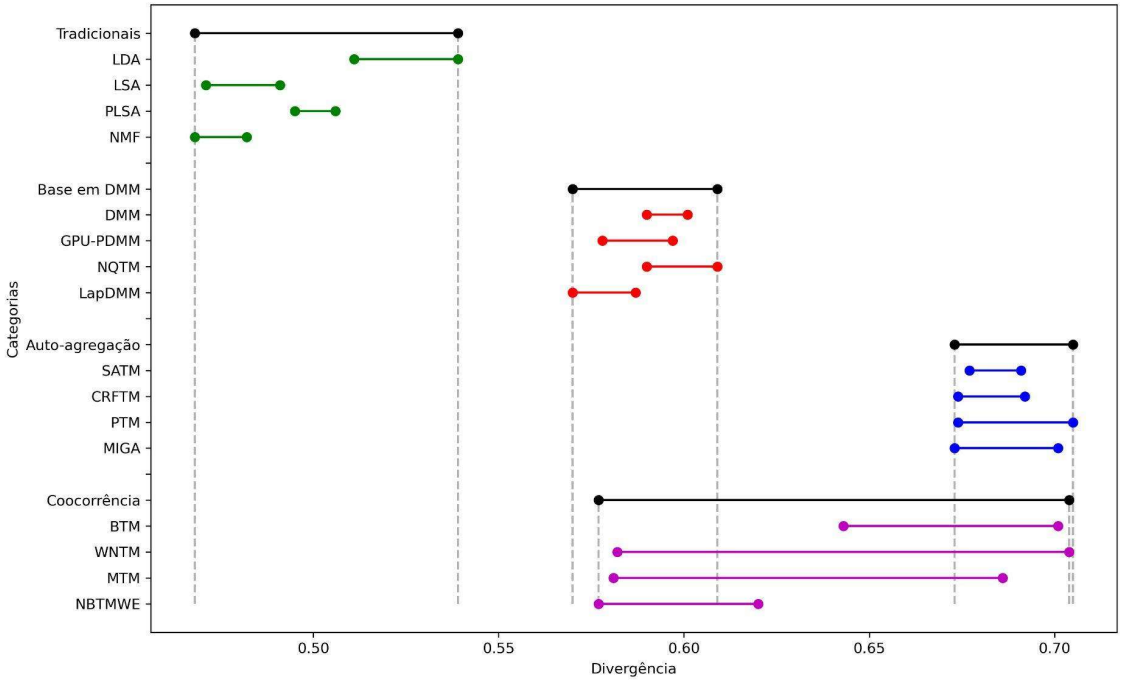
Um valor de divergência-KL simétrica mais alta significa uma maior dissimilaridade entre as distribuições de probabilidade dos tópicos, onde os tópicos são mais distintos e abordam temas diferentes, sem muita sobreposição de termos. Por outro lado, a divergência mais baixa indica uma menor dissimilaridade entre as distribuições, com tópicos compartilhando termos e padrões semelhantes.

Tendências gerais: em geral, os tópicos gerados pelos algoritmos baseados em autoagregação e o BTM (coocorrência) mostraram-se mais distintos, por apresentarem valores mais altos de divergência. Por outro lado, os Tradicionais apresentaram os valores mais baixos de divergência, enquanto os algoritmos baseados em DMM e coocorrência global (exceto BTM) obtiveram valores medianos.

Estabilidade dos algoritmos: em geral, os algoritmos tradicionais e baseados em DMM apresentaram pouca variação individual em seus valores de divergência, conforme mostra a Figura 18.

Apesar do desempenho individual desses algoritmos serem estáveis, os Tradicionais em sua totalidade abrangeram um intervalo maior de divergência, sendo ele 0,46 a 0,54, enquanto os baseados em DMM apresentaram variação de 0,57 a 0,61. Os algoritmos de autoagregação também mostraram estabilidade, com variação de 0,67 a 0,70 embora alguns, como CRFTM e PTM, tiveram uma variação ligeiramente

Figura 18 – Intervalos de divergência-KL simétrica.



mais acentuada que os outros. Por fim, os algoritmos baseados em coocorrência global apresentam maior variação nos valores de divergência, sendo de 0,58 a 0,70. O WNTM e o MTM, em particular, apresentam uma tendência decrescente nos valores de divergência ao longo das iterações.

Comparação entre categorias: os algoritmos Tradicionais apresentaram valores de divergência similares, com LDA mostrando valores ligeiramente mais altos que os outros. Já os algoritmos baseados em DMM, têm valores de divergência um pouco maiores que os algoritmos tradicionais, com o NQTM apresentando os valores mais altos na categoria. Os algoritmos baseados em autoagregação têm valores significativamente mais altos em comparação com as categorias anteriores. Por fim, os algoritmos baseados em coocorrência global apresentam valores de divergência similares aos de autoagregação, mas que variam amplamente, como o BTM e o WNTM, que apresentaram valores mais altos, enquanto o NBTMWE tem valores mais baixos.

Algoritmos com maior divergência de tópicos: em cada categoria, em geral, os algoritmos que apresentaram maior divergência entre tópicos, são:

- Tradicionais: LDA
- Baseados em DMM: NQTM
- Baseados em autoagregação: PTM

- Baseados em coocorrência global: WNTM

Algoritmos com menor divergência de tópicos: há também os algoritmos que apresentaram os menores valores de divergência, sendo eles:

- Tradicionais: LSA, NMF
- Baseados em DMM: LapDMM
- Baseados em autoagregação: CRFTM, PTM, MIGA
- Baseados em coocorrência global: NBTMWE

Em resumo, os algoritmos baseados em autoagregação e o BTM (coocorrência) possuem os valores mais altos, indicando maior dissimilaridade entre os tópicos, enquanto os Tradicionais têm valores menores. O NMF apresenta os menores valores de divergência-KL simétrica entre todos os algoritmos observados, com alta similaridade entre os tópicos gerados. Observou-se também a estabilidade dos algoritmos, com os tradicionais e baseados em DMM mostrando-se mais estáveis, e os baseados em coocorrência global, novamente, com maior sensibilidade às variações do hiperparâmetro ao apresentar grandes flutuações nos valores de divergência-KL simétrica.

6.4 COERÊNCIA C_{UMASS}

A coerência C_{UMASS} é uma medida quantitativa usada para avaliar a qualidade dos tópicos gerados por algoritmos de modelagem de tópicos. Ela se baseia na ideia de que tópicos coerentes devem conter palavras que ocorrem juntas com frequência no corpus. A C_{UMASS} calcula a média do logaritmo das razões entre a coocorrência de pares de palavras do tópico e a frequência de cada palavra individualmente. Essa métrica é especialmente útil para comparar e selecionar modelos de tópicos, ajudando a identificar aqueles que geram tópicos mais interpretáveis e significativos.

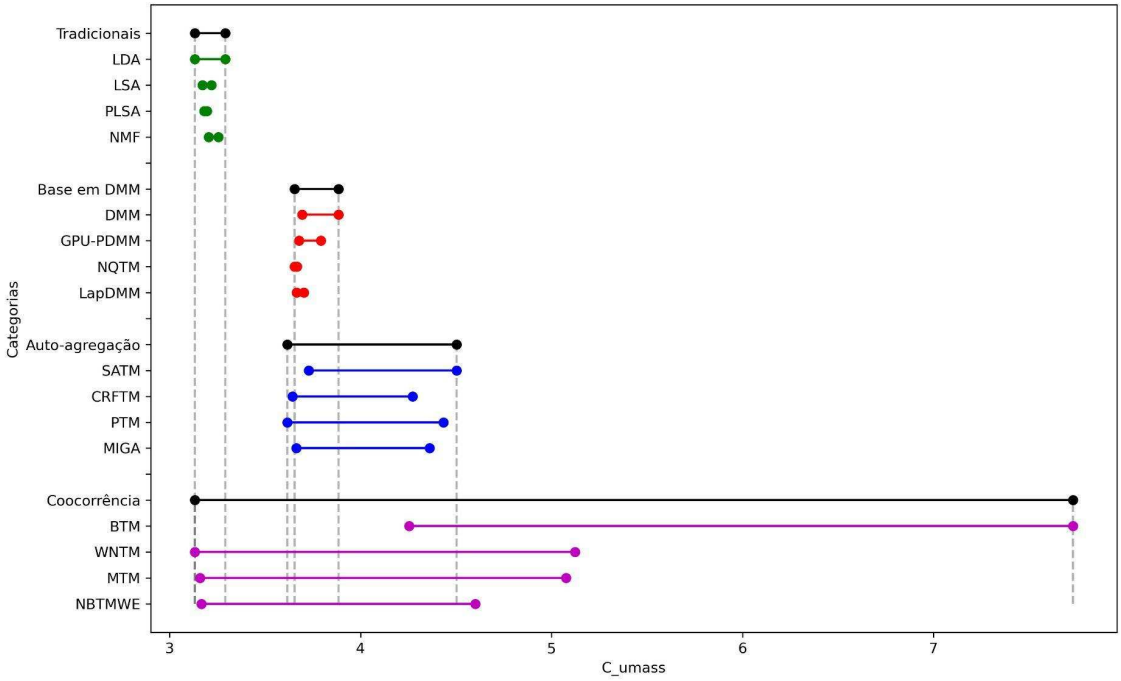
Valores mais altos de C_{UMASS} indicam maior coerência entre as palavras dos tópicos, sugerindo que o algoritmo capturou com sucesso relações semânticas e estruturais relevantes.

Tendências gerais: observou-se que os algoritmos baseados em coocorrência global, especialmente o BTM, apresentaram valores de coerência significativamente mais altos, seguidos pelos algoritmos baseados em autoagregação. Os algoritmos Tradicionais e baseados em DMM têm valores de coerência mais baixos e semelhantes entre si, sendo os Tradicionais ligeiramente menores.

Estabilidade dos algoritmos: os algoritmos Tradicionais e baseados em DMM apresentaram uma estabilidade notável em termos de valores de coerência, com pequenas variações entre diferentes execuções, assim como ilustrado na Figura 19.

As variações estão na faixa de 3,13 a 3,29 para os Tradicionais e 3,65 a 3,88 para os baseados em DMM. Os algoritmos baseados em autoagregação mostram uma tendência inicial de coerência mais alta, porém seguido de uma queda generalizada

Figura 19 – Intervalos de coerência C_{UMASS}.



que representa uma grande variação entre as execuções. Suas variações são de 3,61 a 4,50. Os algoritmos baseados em coocorrência global, por outro lado, exibem uma ampla gama de valores de coerência, indicando uma estabilidade menor e uma sensibilidade maior aos dados e às condições de treinamento, com variações na faixa de 3,13 a 7,73.

Comparação entre categorias: entre os algoritmos tradicionais, o LDA e o NMF exibiram valores de C_{UMASS} um pouco mais altos que o LSA e o PLSA, embora todos os quatro apresentaram valores de coerência relativamente próximos entre si. Na categoria baseada em DMM, o DMM e o GPU-PDMM têm valores de coerência mais altos em comparação com NQTM e LapDMM, porém, a diferença entre os algoritmos dessa categoria é pequena. Os de autoagregação se destacaram por apresentarem valores de coerência significativamente mais elevados em comparação com outras categorias. Especificamente, o SATM e o CRFTM possuem valores de coerência mais altos, enquanto o PTM e o MIGA também apresentam valores mais altos, mas não tão expressivos quanto os dois primeiros. Por fim, a categoria baseada em coocorrência global também apresenta valores de coerência mais elevados em relação aos Tradicionais e baseados em DMM, com destaque para o BTM, que apresentou as maiores coerências nos tópicos gerados nesse estudo.

Algoritmos com maior coerência C_{UMASS}: para cada categoria, foram identificados os algoritmos com maiores C_{UMASS}:

- Tradicionais: NMF
- Baseados em DMM: DMM
- Baseados em autoagregação: SATM
- Baseados em coocorrência global: BTM

Algoritmos com menor coerência C_{UMASS}: também foram identificados, para cada categoria, os algoritmos com menores C_{UMASS}:

- Tradicionais: LDA
- Baseados em DMM: NQTM, LapDMM
- Baseados em autoagregação: PTM
- Baseados em coocorrência global: WNTM, MTM, NBTMWE

Em resumo, os algoritmos baseados em coocorrência global foram os mais coerentes conforme a C_{UMASS}, seguidos pelos algoritmos baseados em autoagregação, enquanto os algoritmos tradicionais e baseados em DMM apresentaram desempenho semelhantes e menores. A estabilidade dos algoritmos variou, sendo que os algoritmos baseados em coocorrência global foram, mais uma vez, os mais sensíveis, enquanto os tradicionais e baseados em DMM foram mais estáveis.

O BTM, em particular, se destacou nessa avaliação. Isso pode ser explicado, pois, essencialmente, ele usa os pares de palavras nos documentos para formar os tópicos (CHENG *et al.*, 2014). Enquanto o C_{UMASS} mede o quanto os tópicos foram formados usando palavras que eram próximas de si no corpus. Os demais algoritmos de coocorrência possuem abordagens similares, contudo possuem outras soluções em suas composições que afetaram os resultados.

6.5 COERÊNCIA C_{W2V}

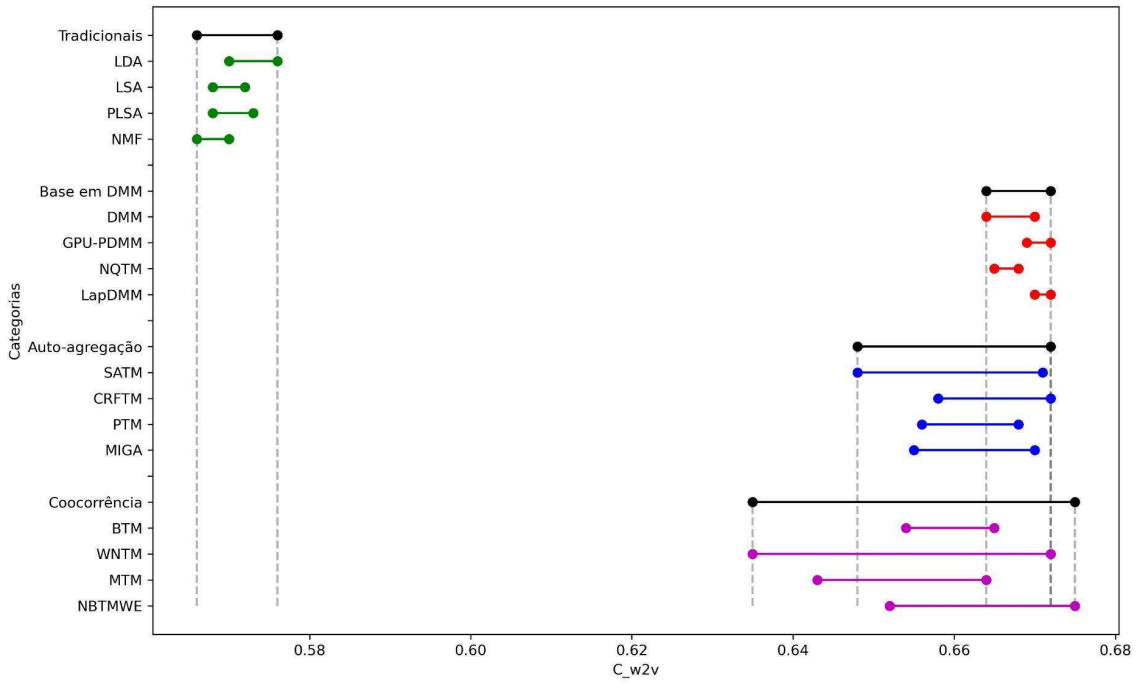
A métrica de coerência C_{W2V} é uma medida de coerência baseada no *Word2Vec*, uma técnica de *deep learning* que gera representações vetoriais de palavras, considerando o contexto em que elas aparecem. No contexto da modelagem de tópicos, a coerência C_{W2V} quantifica a semelhança semântica entre as palavras mais representativas de um tópico, calculando a média das similaridades de cosseno entre os vetores *Word2Vec* das palavras.

Tópicos com valores de C_{W2V} mais altos são considerados mais coerentes e significativos, pois suas palavras representativas compartilham relações semânticas mais fortes, enquanto tópicos com valores mais baixos de coerência C_{W2V} são menos coerentes e podem ser mais difíceis de interpretar.

Tendências gerais: observou-se que os tópicos gerados pelos algoritmos baseados em DMM e autoagregação tenderam a apresentar mais coerência e relevância semântica, com valores mais altos de C_{W2V}, enquanto os algoritmos tradicionais e baseados em coocorrência mostram, em geral, valores mais baixos.

Estabilidade dos algoritmos: em geral, todos os algoritmos apresentaram uma certa estabilidade em seus resultados, com variações baixas nos resultados mesmo com as mudanças no número de tópicos, assim como mostra a Figura 20.

Figura 20 – Intervalos de coerência C_{W2V}.



No entanto, algoritmos como o WNTM e NBTMWE, apresentaram uma variação ligeiramente mais acentuada nos resultados, mostrando uma instabilidade em comparação com outros algoritmos. De forma geral, os Tradicionais variaram de 0,57 a 0,58, os baseados em DMM ficaram na faixa de 0,66 a 0,67, os de autoagregação foram de 0,64 a 0,67, enquanto os de coocorrência global apresentaram variação de 0,64 a 0,68.

Comparação entre categorias: os algoritmos tradicionais apresentaram valores de coerência semelhantes entre si e relativamente baixos. Os baseados em DMM apresentaram valores de coerência mais altos do que os algoritmos tradicionais e baseados em coocorrência. Entre eles, o GPU-PDMM e o LapDMM parecem ter valores ligeiramente mais altos em comparação aos outros. Quantos aos baseados em autoa-

gregação, também apresentaram valores de coerência mais altos do que os algoritmos tradicionais e baseados em coocorrência. Nessa categoria, CRFTM apresenta os valores de coerência mais altos. E os baseados em coocorrência, exibem valores de C_{W2V} variados, mas geralmente mais baixos em comparação com os algoritmos baseados em DMM e autoagregação.

Algoritmos com maior coerência C_{W2V}: dentro de cada categoria, os algoritmos que apresentaram maiores C_{W2V} são:

- Tradicionais: LDA
- Baseados em DMM: GPU-PDMM, LapDMM
- Baseados em autoagregação: SATM, CRFTM
- Baseados em coocorrência global: NBTMWE

Algoritmos com menor coerência C_{W2V}: também foram identificados, para cada categoria, os algoritmos com menores C_{W2V}:

- Tradicionais: NMF
- Baseados em DMM: DMM, NQTM
- Baseados em autoagregação: SATM
- Baseados em coocorrência global: WNTM

Em resumo, a análise da coerência C_{W2V} sobre esses algoritmos mostrou que, de maneira geral, os algoritmos baseados em DMM e autoagregação apresentaram desempenho superior em comparação aos algoritmos tradicionais e baseados em coocorrência, gerando tópicos com maior coerência C_{W2V}. Na comparação entre as categorias, os algoritmos baseados em DMM e autoagregação se destacaram, apresentando valores de coerência superiores aos demais. Além disso, dentro de cada categoria, existem algumas diferenças de desempenho entre os algoritmos individualmente, mas, em geral, essas diferenças são pequenas. Todos os algoritmos apresentaram uma variação relativamente baixa nos valores de coerência, indicando que os resultados são consistentes entre diferentes configurações de parâmetros.

Os resultados obtidos com a C_{W2V} foram os mais estáveis e semelhantes entre si. Essa similaridade pode ser atribuída ao fato de que o C_{W2V} utiliza vetores de palavras pré-treinados, capturando assim informações semânticas globais (O'CALLAGHAN *et al.*, 2015). Como resultado, essa coerência pode ser mais tolerante às diferenças específicas de cada algoritmo, fornecendo uma visão mais unificada entre os desempenhos de algoritmos de uma mesma categoria, que compartilham técnicas comuns entre todos para modelar os tópicos.

6.6 DISTÂNCIA E DIVERGÊNCIA

Este estudo apresenta uma apuração dos desempenhos de diversos algoritmos de modelagem de tópicos ao serem aplicados em textos curtos advindos de mídias sociais. Para tanto, duas das cinco métricas utilizadas para realizar as medições são relacionadas à similaridade que os resultados dos algoritmos apresentam. Nesse contexto, é possível verificar o quão apropriada determinada técnica é, ao medir a similaridade dos tópicos gerados por um algoritmo.

Assim, a distância dos tópicos é usada como medida que quantifica o quão distintos são os tópicos gerados pelos algoritmos. Valores menores indicam que os tópicos são mais similares entre si, enquanto valores maiores indicam uma maior diferenciação entre os tópicos. Portanto, a distância de tópicos refere-se à capacidade de um algoritmo de distinguir e separar tópicos distintos uns dos outros.

Já a divergência-KL simétrica pode medir a qualidade da modelagem de tópicos. A divergência quantifica a dissimilaridade entre duas distribuições de probabilidade (neste caso, tópicos). Valores menores de divergência indicam que as distribuições são mais semelhantes e, consequentemente, os tópicos gerados não são bem diferenciados, levando a uma sobreposição de conteúdo e menos distinção entre os tópicos. Por outro lado, valores maiores indicam uma maior diferença entre as distribuições, com tópicos mais específicos e bem definidos. Isso implica que os documentos associados a esses tópicos compartilham características comuns e são mais facilmente agrupados, o que geralmente é desejável em uma análise de tópicos.

Um algoritmo, por exemplo, pode apresentar um valor baixo de divergência e um valor maior para distância, em um caso específico. Nesse caso, o algoritmo geraria tópicos com distribuições de probabilidade muito semelhantes entre si, e, mesmo assim, teria tópicos bem separados no espaço de características. Isso não significa necessariamente que o algoritmo é bom ou ruim, mas indica que o algoritmo está capturando um aspecto específico dos dados e não está modelando a variação entre os tópicos idealmente.

Por isso, ao avaliar um algoritmo de modelagem de tópicos, é importante considerar várias métricas e não basear a avaliação em uma única métrica. Além disso, é útil comparar os resultados do algoritmo em questão com os de outros algoritmos para obter uma compreensão mais completa de seu desempenho.

Nesse sentido, os algoritmos Tradicionais apresentaram valores relativamente baixos de divergência-KL simétrica, indicando a limitação deles em gerar tópicos com alta distribuição de probabilidade quando aplicados a textos curtos. A distância dos tópicos dos Tradicionais também não é tão alta, mostrando uma diferenciação de tópicos baixa a moderada.

Os algoritmos baseados em DMM mostraram divergências mais altas do que os Tradicionais, indicando uma maior capacidade dessa categoria em lidar com as

distribuições de probabilidade dos textos curtos. Apesar disso, em média, a distância de tópicos para esses algoritmos também não foi muito alta, mas ainda mostraram uma diferenciação um pouco maior que os Tradicionais. Isso indica que, nesse contexto, os algoritmos baseados em DMM são uma alternativa viável para pesquisas e estudos que envolvam análise de subtemas.

Os baseados em autoagregação apresentaram divergências-KL simétrica ainda mais altas em comparação aos Tradicionais. Eles mostraram também distâncias de tópicos bastante altas, obtendo um bom equilíbrio entre as duas avaliações. Isso sugere que os algoritmos dessa categoria são um ponto mais equilibrado entre similaridade e diferença de tópicos.

Por fim, os baseados em coocorrência global de palavras. Esses apresentaram uma mistura de resultados para a divergência e a distância dos tópicos. Em geral, a distância e divergência desses algoritmos se assemelha aos baseados em autoagregação, com divergência inicialmente mais alta que diminui conforme se aumenta o número de tópicos. Porém, o BTM é o único que se mantém relativamente estável, apresentando menor queda no desempenho. Isso indica que os baseados em coocorrência global, a depender das configurações usadas, também são um ponto de equilíbrio entre similaridade e diferença. Todavia, o BTM se distingue nessa categoria. Esse algoritmo mostrou resultados consistentemente maiores em boa parte dos testes, mesmo quando os outros apresentavam uma queda brusca no desempenho. Portanto, pode-se dizer que, dentre todos os algoritmos avaliados nesse trabalho, o BTM é o mais adequado para realizar uma análise exploratória em textos advindos de mídias sociais.

6.7 ALGORITMOS TRADICIONAIS VS. MODERNOS

De acordo com os resultados mostrados anteriormente, a Tabela 3 reúne os métodos que apresentaram os menores e maiores valores em cada métrica.

Os algoritmos Tradicionais (LDA, LSA, PLSA, NMF) apresentam consistentemente os resultados mais altos de perplexidade e mais baixos para as outras métricas. Isso reflete a incapacidade desses em se adequarem aos dados de textos utilizados. Considerando os resultados obtidos e as características dos textos utilizados neste trabalho, uma modelagem de tópicos realizada por algum desses métodos Tradicionais até pode resultar em conjuntos com palavras pertinentes ao tema central (COVID-19), porém é evidente que há um prejuízo nesse produto. Com perplexidade mais alta e coerências (C_{UMASS}, C_{W2V}) abaixo da média, é notório que esses resultados estão aquém do que outras alternativas podem alcançar.

Para fins de comparação, a Tabela 4 mostra exemplos de tópicos gerados pelo LDA (Tradicional), o DMM (baseados em DMM), o SATM (autoagregação), e pelo BTM (coocorrência), a partir do mesmo corpus. Esses tópicos são relativos a modelagens

Tabela 3 – Listagem dos algoritmos com maiores e menores resultados para cada métrica em geral.

Algoritmo	**Métrica**				
	Perplexidade	**Distância**	**Divergência**	C_{UMASS}	C_{W2V}
LDA		MENOR		MENOR	
LSA			MENOR		
PLSA		MENOR			
NMF	MAIOR		MENOR		MENOR
PTM			MAIOR		
BTM				MAIOR	
WNTM		MAIOR	MAIOR	MENOR	
MTM	MENOR			MENOR	
NBTMWE	MENOR			MENOR	MAIOR

Tabela 4 – Exemplos de tópicos gerados pelo LDA e BTM.

Algoritmos	**Tópico**	**Palavras**									
LDA	**1**	covid19	coronavirus	caso	brasil	pessoa	morte	saude	pandemia	casa	morrer
	2	coronavirus	covid19	brasil	caso	pessoa	bolsonaro	morte	presidente	saber	saude
	3	covid19	coronavirus	morte	caso	brasil	bolsonaro	falar	saude	pessoa	morrer
DMM	**1**	coronavirus	chegar	bolsonaro	matar	cancelar	mundo	causa	medo	pessoa	novo
	2	virus	corona	matar	bolsonaro	longe	demais	pneumonia	china	passar	casos
	3	corona	confirmar	dois	novo	caso	morrer	rio	mundo	italia	china
SATM	**1**	coronavirus	covid19	pessoa	brasil	virus	ficar	dia	gente	casa	falar
	2	coronavirus	bolsonaro	casos	brasil	influencer	pegar	falar	mundo	dia	pessoas
	3	bolsonaro	irresponsavel	bbb20	globo	ficar	casa	pessoa	corona	medo	gripe
BTM	**1**	ficar	casa	sair	pessoa	quarentena	medo	bem	longe	doente	mãe
	2	brasil	primeiro	infectado	chegar	italia	morte	número	subir	presidente	nada
	3	vídeo	cardi	falar	coronavirus	pai	mãe	ouvir	preocupar	dia	aula

onde $K = 10$, sendo que três resultados de cada uma foram escolhidos para exibição na tabela.

Conforme mostra a Tabela 4, os termos presentes nos tópicos do LDA se repetem com muita frequência, especialmente “covid19”, “coronavirus”, “brasil”, e “morte”. Isso significa que as mesmas palavras compõem quase metade de vários tópicos do LDA. Um motivo crucial para isso acontecer é a frequência dessas palavras. Para ilustrar melhor porque isso acontece, a Figura 21 ilustra uma nuvem de palavras, com os 50 termos mais frequentes em todo o corpus.

Tendo em vista as limitações do LDA em lidar com textos curtos, e de outros algoritmos Tradicionais, é evidente que palavras com alta frequência foram “beneficiadas” nas modelagens. A palavra “coronavirus”, por exemplo, apareceu em cerca de 30% dos quase 4 milhões de documentos, enquanto “covid19” apareceu em aproximadamente 15% dos textos. Além do mais, ressalta-se que todos os textos do corpus são formados por um total de 37.463.729 palavras, sendo que as palavras presentes na nuvem da Figura 21 representam 22,51% dessas palavras.

O DMM e o SATM apresentaram resultados semelhantes. Eles possuem um

Figura 21 – Os 50 termos mais frequentes em todo o corpus.

certo nível de semelhança entre os tópicos, mas cada um também tem suas particularidades que mostram como os resultados desses algoritmos são equilibrados entre identificar padrões e correlações emergentes no corpus, e extrair detalhes mais profundos e refinados a partir de um tema principal já reconhecido.

O BTM, por sua vez, mostrou uma diversidade bem maior de palavras, além de apresentar mais coerência entre elas. Como pode ser visto na Tabela 4, o assunto do tópico 1 envolve quarentena, pessoas ficando em casa, e o medo que elas sentiam. O tópico 2 trata do primeiro infectado no Brasil, e o número de mortes subindo na Itália. Já o terceiro tópico abrange, entre outros assuntos, o vídeo viral da artista estadunidense Cardi B, no qual ela expressa calorosamente seu medo com o emergente surto da COVID-19 em 2020. Por fim, é importante ressaltar que, nesse caso, devido ao alto número de documentos usados e o baixo valor de K, os tópicos foram gerados englobando múltiplos assuntos.

6.8 CONSIDERAÇÕES GERAIS

No presente capítulo, é feita uma ampla análise e discussão sobre os resultados expostos no capítulo 6. Os resultados foram examinados mais amplamente, abrangendo todas as categorias de algoritmos, para identificar quais categorias apresentaram os maiores e menores desempenhos em cada métrica.

Há também uma comparação entre as métricas de distância de tópicos e divergência-KL simétrica. As duas possuem um propósito semelhante: medir a similaridade dos tópicos gerados numa modelagem. Devido a isso, é necessário ressaltar que uma não exclui a outra. Pelo contrário, quando usadas em conjunto para avaliar os algoritmos, é possível ter um entendimento ainda mais amplo sobre a capacidade de cada técnica.

Sobre os resultados, em geral, os algoritmos baseados em coocorrência global apresentam os menores resultados de perplexidade, seguidos pelos algoritmos baseados em autoagregação. Os Tradicionais apresentaram a menor capacidade de generalização, apesar disso o LDA se distingue entre eles e apresenta valores mais razoáveis.

Com relação à distância de tópicos e divergência-KL simétrica, os baseados em autoagregação e coocorrência global mostraram desempenho maior, enquanto os Tradicionais apresentaram os menores valores.

E quanto as coerências C_{UMASS} e C_{W2V}, os algoritmos de modelagem de tópicos baseados em coocorrência global apresentaram, em média, o maior desempenho, com destaque para o BTM. Todavia, os de autoagregação e DMM seguem ligeiramente abaixo, com pouca diferença entre essas três categorias. Os Tradicionais novamente se mostraram os menores resultados.

Observou-se nos resultados que, na maioria das avaliações, o comportamento entre algoritmos de uma mesma categoria apresentou performances similares, com alguns casos ficando bastante próximos.

Com isso, destaca-se que a escolha de uma técnica específica utilizada pela categoria (ex: autoagregação de documentos, ou coocorrência global de palavras) mostra-se mais impactante para o resultado do que a escolha de um algoritmo individualmente. Além disso, a escolha da solução mais adequada para a modelagem de tópicos depende do objetivo e das prioridades de sua aplicação, pois dependendo da aplicação e dos requisitos específicos, pode ser mais relevante priorizar uma competência da solução em detrimento da outra. É importante considerar o propósito das métricas de avaliação e pesar a importância relativa de cada uma para decidir qual usar.

Isso posto, ao considerar as métricas de avaliação utilizadas, e como essas mensuram as capacidades de cada algoritmo, a análise de subtemas e a análise exploratória se apresentaram como lados opostos no espectro da modelagem de tópicos. Portanto, considerando esses dois polos, e para melhor compreensão dos resultados obtidos, os algoritmos são referidos de acordo com seus polos mais convenientes.

Os resultados obtidos na avaliação realizada neste trabalho mostraram que os algoritmos baseados em DMM tem maior potencial para modelar tópicos a partir de textos curtos visando analisar subtemas presentes no corpus. A capacidade de gene-

ralização desses algoritmos mostrou-se inferior aos de autoagregação e coocorrência global em alguns casos, mas o NQTM, por exemplo, ainda mostrou um resultado razoável. A distância dos tópicos, em geral, mostrou-se mais baixa, mas não tanto quanto os Tradicionais, e por isso os tópicos gerados tendem a manter sempre palavras-chave sobre o tema a ser analisado. Na divergência, este se encontra ligeiramente abaixo dos baseados em autoagregação e coocorrência global, mas ainda é um valor razoável, diferente dos Tradicionais que estão ainda mais baixos. E quanto as coerências C_{UMASS} e C_{W2V}, observa-se resultados próximos entre as categorias de baseados em DMM, autoagregação, e coocorrência global. A única categoria evidentemente abaixo das outras é a de algoritmos Tradicionais.

Ademais, o algoritmo BTM, de coocorrência global, destacou-se em sua categoria, mostrando resultados consistentemente bastante acima dos outros em distância, divergência, e coerência C_{UMASS}. Isso indica que esse algoritmo pode realizar com mais eficiência uma análise exploratória em textos curtos de mídias sociais. Já as categorias de autoagregação e coocorrência global, exceto o BTM, se mostraram mais equilibradas entre essas duas extremidades. E por fim, os algoritmos Tradicionais apresentaram os menores desempenhos em quase todas as avaliações, com perplexidades altas, e distâncias, divergências, e coerências mais baixas que os demais. Apesar desses resultados, o LDA se destacou de forma que, a depender do caso específico e das características do conjunto de dados em questão, pode ser uma abordagem válida para investigar subtemas em um corpus de textos.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 REVISÃO DAS MOTIVAÇÕES E OBJETIVOS

A popularização das mídias sociais tornaram-nas uma ferramenta indispensável para a participação em discussões online e a contribuição com opiniões e informações. Entretanto, a enorme quantidade de dados gerados e a predominância de mensagens curtas nas plataformas online dificultam a identificação de tendências relevantes e significativas nas discussões.

Assim surge a modelagem de tópicos, uma técnica de PLN que permite identificar os grupos de palavras mais relevantes em um conjunto de textos, contribuindo para a compreensão dos temas e padrões latentes nas conversas online. Dessa forma, essa técnica possibilita a identificação de tendências e influências que podem afetar atitudes e comportamentos individuais.

Esse trabalho realizou um levantamento sobre as técnicas de modelagem de tópicos usadas em textos curtos, onde se observou que muitos trabalhos utilizam abordagens clássicas para lidar com textos coletados nas mídias sociais. Todavia, isso pode ser um problema, pois tais abordagens foram feitas para textos mais longos, numa época onde não se encontravam textos curtos com tanta abundância como nas mídias sociais.

Esse trabalho visa, entre outras coisas, compreender se estas práticas são realmente adequadas para esse tipo de texto. O objetivo geral desse trabalho foi demonstrar o desempenho de abordagens tradicionais de modelagem de tópicos em relação a outras categorias mais especializadas para a aplicação em uma coleção de textos provenientes de mídias sociais e identificar como cada categoria de algoritmo performou em um cenário real.

Para isso, objetivos específicos foram seguidos. Foi necessário um levantamento bibliográfico para identificar e reunir uma seleção de técnicas de modelagem de tópicos para textos curtos e métricas para poder avaliá-los. Além disso, um corpus de textos com discussões reais foi produzido, advindo do Twitter. Os dados brutos de texto passaram por uma etapa de preparação, onde uma amostra de aproximadamente 4 milhões de textos foram selecionados e pré-processados. Então, a modelagem de tópicos foi executada com 16 algoritmos distintos e cada modelo foi avaliado a partir de 5 métricas. Por fim, os resultados obtidos foram expostos e discutidos.

7.2 VISÃO GERAL DO TRABALHO

Nesta seção é feita uma revisão do trabalho realizado e de como foram atendidos os objetivos listados anteriormente.

O capítulo 1 desta dissertação descreveu o contexto em que o trabalho está

inserido, o objetivo geral e os objetivos específicos. Os capítulos 2, 3, e 4 envolvem a revisão da literatura. O capítulo 2 trata da fundamentação teórica e trouxe os conceitos utilizados no restante do trabalho. No capítulo 3 foram apresentados os algoritmos de modelagem de tópicos avaliados nesse trabalho, separando-os em quatro categorias e mostrando os conceitos básicos de operação de cada um deles. O capítulo 4 apresentou as métricas de avaliação comumente utilizadas para medir o desempenho de algoritmos. O capítulo 5 mostra a metodologia adotada para a realização deste estudo. O capítulo 6 apresentou uma análise comparativa da performance dos algoritmos. Os resultados foram exibidos integralmente no Apêndice A.

Além disso, constatou-se que os algoritmos Tradicionais apresentaram desempenho claramente inferiores, evidenciando os cuidados que se deve ter ao utilizá-los para textos curtos. Já os algoritmos baseados em DMM se mostraram mais viáveis para análises de subtemas. Por outro lado, o BTM, especificamente, se mostrou mais adequado para análises exploratórias dos dados, enquanto os baseados em autoagregação e coocorrência global demonstraram resultados mais balanceados.

7.3 CONTRIBUIÇÕES

Segundo os objetivos definidos para este trabalho, pode-se listar as seguintes contribuições:

1. Uma seleção de técnicas de modelagem de tópicos para serem aplicadas em textos curtos, e de métricas para avaliar o desempenho desses algoritmos;
2. Um fluxo metodológico para modelagem de tópicos em textos originados em mídias sociais;
3. A implementação dos algoritmos selecionados, incluindo otimizações de paralelização e uso de matrizes esparsas;
4. Uma análise comparativa dos desempenhos dos algoritmos quando aplicados a textos curtos reais, usando as métricas selecionadas.

Visando divulgar os resultados obtidos com o trabalho e, principalmente, submetê-los para uma avaliação da comunidade científica, um artigo foi produzido e submetido a um evento nas áreas de computação de *data mining* e descoberta de conhecimento. As revisões e discussões desta publicação contribuíram para o amadurecimento deste trabalho, incluindo melhorias no tratamento dos dados e avaliações de desempenho. O artigo publicado está listado a seguir:

1. SANTOS, Ian; RECH, Luciana; MORAES, Ricardo. A Topic Modeling Method for Analysis of Short-Text Data in Social Media Networks. In: Proceedings of 37th International Conference on Computers and Their Applications - CATA 2022. [s.l.], 2022 v. 82, p. 112-121. Qualis **B3**.

Outro artigo encontra-se em fase de submissão e apresenta a análise de desempenho descrita no capítulo 6, relatando as descobertas mais recentes deste estudo.

7.4 LIMITAÇÕES

Embora este trabalho tenha atingido os objetivos desejados, algumas decisões tomadas em seu desenvolvimento trouxeram limitações à sua utilização. Essas limitações são discutidas a seguir.

Este trabalho apresentou uma comparação de desempenho entre diversos algoritmos, contudo, a única variação de hiperparâmetros testada para os modelos foi K, o número de tópicos. Outros parâmetros foram mantidos o mais próximo possível do que é considerado valor padrão, de acordo com seus respectivos autores. Isso acontece, pois, nem sempre os 16 algoritmos compartilharam os mesmos conjuntos de hiperparâmetros configuráveis. A diversidade de abordagens envolveu diferentes técnicas de computação, desde grafos (que montam redes de palavras) até o uso de aprendizado de máquina. Apesar disso, sempre que possível, as configurações foram mantidas semelhantes.

Os testes nesse trabalho foram realizados com apenas um corpus de textos. Essa estratégia foi necessária para concentrar os recursos disponíveis para essa pesquisa de forma que o resultado fosse mais profundo e focado, maximizando o uso dos meios disponíveis. Da mesma forma, ao focar em um único conjunto de dados, foi possível garantir uma maior consistência e comparabilidade dos resultados. Isso permitiu uma análise mais direta e menos sujeita a variabilidades inerentes a conjuntos de textos distintos, além de tornar a validação interna dos resultados mais robusta. Ressalta-se também que, trabalhos similares na literatura, em outros contextos de modelagem de tópicos, mostraram que testes em diferentes corpus não apresentam grande variação na média final dos resultados (RÜDIGER *et al.*, 2022; VAYANSKY; KUMAR, 2020).

Por fim, é importante salientar que não existe, na comunidade científica, um consenso sobre quais métricas de avaliação são essenciais para medir o desempenho de uma modelagem de tópicos convencional, muito menos para uma em textos curtos. Portanto, para esse estudo, elas foram escolhidas de acordo com aspectos considerados importantes após ampla pesquisa sobre o assunto. A adoção dessas métricas por outros trabalhos também foi um fator relevante.

7.5 TRABALHOS FUTUROS

Como trabalhos futuros inclui-se uma investigação do impacto que um número de tópicos ainda mais amplo poderia representar. Contudo, o aumento do número de tópicos gerados numa modelagem leva, em todos os casos, ao crescimento da

complexidade computacional, portanto é necessário que a programação envolvida seja otimizada para isso e os equipamentos usados consigam suportar essa carga maior.

É válido também investigar como as diferentes configurações de hiperparâmetros afetam o desempenho dos algoritmos. Cada algoritmo de modelagem de tópicos tem seu próprio conjunto de hiperparâmetros que podem ser ajustados, e seria pertinente investigar o impacto desses hiperparâmetros no desempenho do algoritmo. Da mesma forma, a realização dos testes em outros cenários, outros idiomas, e com quantidades diferentes de documentos sendo usados na modelagem, mostrará como essas soluções desempenham ao tratar de menos, ou mais, documentos.

A discussão do trabalho pode ser ampliada, principalmente, ao abranger outros algoritmos de modelagem de tópicos e/ou métricas de avaliação. É impraticável que um trabalho apenas consiga englobar todas as soluções existentes, mas seguindo uma metodologia robusta, é possível criar uma maneira de medir outros algoritmos já existentes, e até os que ainda virão a existir.

Um aspecto muito importante na modelagem de tópicos são os tópicos em si. É extremamente relevante medir a interpretabilidade dos grupos de palavras gerados nas modelagens. Até o momento, não existe uma métrica de avaliação automatizada para realizar essa atividade tão eficientemente quanto a intuição de um avaliador humano.

Por fim, seguindo por outra direção, seria proveitosa a integração dos algoritmos de modelagem de tópicos com outras técnicas de PLN, como:

- Análise de sentimentos, para identificar as emoções ou sentimentos associados a cada tópico;
- Reconhecimento de entidades nomeadas, para uma melhor compreensão do contexto e das entidades em torno de cada tópico;
- Sumarização de textos, combinado com a modelagem de tópicos pode criar sumários informativos que também indicam os principais assuntos discutidos no corpus;
- Classificação de texto, onde a modelagem de tópicos pode ser usada para melhorar a performance da classificação dos textos ao fornecer características adicionais para o modelo de classificação.

REFERÊNCIAS

AGRAWAL, Rakesh; IMIELIŃSKI, Tomasz; SWAMI, Arun. Mining association rules between sets of items in large databases. *In*: PROCEEDINGS of the 1993 ACM SIGMOD international conference on Management of data. [*S.l.*: *s.n.*], 1993. P. 207–216.

AHMAD, Amir; KHAN, Shehroz S. Survey of state-of-the-art mixed data clustering algorithms. **Ieee Access**, IEEE, v. 7, p. 31883–31902, 2019.

ALASHRI, Saud; KANDALA, Srinivasa Srivatsav; BAJAJ, Vikash; RAVI, Roopek; SMITH, Kendra L; DESOUZA, Kevin C. An analysis of sentiments on facebook during the 2016 US presidential election. *In*: IEEE. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). [*S.l.*: *s.n.*], 2016. P. 795–802.

ALGHAMDI, Rubayyi; ALFALQI, Khalid. A survey of topic modeling in text mining. **Int. J. Adv. Comput. Sci. Appl.(IJACSA)**, Citeseer, v. 6, n. 1, 2015.

ALSUMAIT, Loulwah; BARBARÁ, Daniel; GENTLE, James; DOMENICONI, Carlotta. Topic Significance Ranking of LDA Generative Models. *In*: BUNTINE, Wray; GROBELNIK, Marko; MLADENIĆ, Dunja; SHAWE-TAYLOR, John (Ed.). **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. P. 67–82.

ANAND, Rajaraman; JEFFREY DAVID, Ullman. **Mining of massive datasets**. [*S.l.*]: Cambridge university press, 2011.

ARUN, R.; SURESH, V.; VENI MADHAVAN, C. E.; NARASIMHA MURTHY, M. N. On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. *In*: ZAKI, Mohammed J.; YU, Jeffrey Xu; RAVINDRAN, B.; PUDI, Vikram (Ed.). **Advances in Knowledge Discovery and Data Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. P. 391–402.

AWASTHI, Ishitva; GUPTA, Kuntal; BHOGAL, Prabjot Singh; ANAND, Sahejpreet Singh; SONI, Piyush Kumar. Natural language processing (NLP) based text summarization-a survey. *In*: IEEE. 2021 6th International Conference on Inventive Computation Technologies (ICICT). [*S.l.*: *s.n.*], 2021. P. 1310–1317.

BANDA, Juan M.; TEKUMALLA, Ramya; WANG, Guanyu; YU, Jingyuan; LIU, Tuo; DING, Yuning; ARTEMOVA, Ekaterina; TUTUBALINA, Elena; CHOWELL, Gerardo. A Large-Scale COVID-19 Twitter Chatter Dataset for Open Scientific Research—An International Collaboration. **Epidemiologia**, v. 2, n. 3, p. 315–324, 2021.

BARDE, Bhagyashree Vyankatrao; BAINWAD, Anant Madhavrao. An overview of topic modeling methods and tools. *In*: IEEE. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS). [*S.l.*: *s.n.*], 2017. P. 745–750.

BELFORD, Mark; MAC NAMEE, Brian; GREENE, Derek. Stability of topic modeling via matrix factorization. **Expert Systems with Applications**, v. 91, p. 159–169, 2018. ISSN 0957-4174.

BISANDU, Desmond Bala; PRASAD, Rajesh; LIMAN, Musa Muhammad. Clustering news articles using efficient similarity measure and N-grams. **International Journal of Knowledge Engineering and Data Mining**, Inderscience Publishers (IEL), v. 5, n. 4, p. 333–348, 2018.

BLEI, David M. Probabilistic topic models. **Communications of the ACM**, ACM New York, NY, USA, v. 55, n. 4, p. 77–84, 2012.

BLEI, David M; LAFFERTY, John D. Dynamic topic models. *In*: PROCEEDINGS of the 23rd international conference on Machine learning. [*S.l.*: *s.n.*], 2006. P. 113–120.

BLEI, David M; NG, Andrew Y; JORDAN, Michael I. Latent dirichlet allocation. **Journal of machine Learning research**, v. 3, Jan, p. 993–1022, 2003.

CAO, Juan; XIA, Tian; LI, Jintao; ZHANG, Yongdong; TANG, Sheng. A density-based method for adaptive LDA model selection. **Neurocomputing**, v. 72, n. 7, p. 1775–1781, 2009. Advances in Machine Learning and Computational Intelligence.

CARON, Mathilde; BOJANOWSKI, Piotr; JOULIN, Armand; DOUZE, Matthijs. Deep clustering for unsupervised learning of visual features. *In*: PROCEEDINGS of the European conference on computer vision (ECCV). [*S.l.*: *s.n.*], 2018. P. 132–149.

CHANDRA, SS; HAREENDRAN, S *et al.* **Machine Learning: A Practitioner's Approach**. [*S.l.*]: PHI Learning Pvt. Ltd., 2021.

CHANG, Jonathan; GERRISH, Sean; WANG, Chong; BOYD-GRABER, Jordan; BLEI, David. Reading Tea Leaves: How Humans Interpret Topic Models. *In*: BENGIO, Y.; SCHUURMANS, D.; LAFFERTY, J.; WILLIAMS, C.; CULOTTA, A. (Ed.). **Advances in Neural Information Processing Systems**. [*S.l.*]: Curran Associates, Inc., 2009. v. 22.

CHEN, Po-Hao; ZAFAR, Hanna; GALPERIN-AIZENBERG, Maya; COOK, Tessa. Integrating natural language processing and machine learning algorithms to categorize oncologic response in radiology reports. **Journal of digital imaging**, Springer, v. 31, p. 178–184, 2018.

CHEN, May-Ru; KUBA, Markus. On Generalized Pólya Urn Models. **Journal of Applied Probability**, Cambridge University Press, v. 50, n. 4, p. 1169–1186, 2013.

CHENG, Xueqi; YAN, Xiaohui; LAN, Yanyan; GUO, Jiafeng. Btm: Topic modeling over short texts. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 26, n. 12, p. 2928–2941, 2014.

CHOWDHARY, KR. **Fundamentals of artificial intelligence**. [*S.l.*]: Springer, 2020.

CHURCHILL, Rob; SINGH, Lisa. The Evolution of Topic Modeling. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 54, 10s, nov. 2022.

COPPERSMITH, Glen; LEARY, Ryan; CRUTCHLEY, Patrick; FINE, Alex. Natural language processing of social media as screening for suicide risk. **Biomedical informatics insights**, SAGE Publications Sage UK: London, England, v. 10, 2018.

DAS, Arijita; JANA, Soumita; GANGULY, Pranita; CHAKRABORTY, Nirban. Application of Association Rule: Apriori Algorithm in E-Commerce. *In*: IEEE. 2021 Innovations in Energy Management and Renewable Resources (52042). [*S.l.*: *s.n.*], 2021. P. 1–7.

DE LUCCA, JL; NUNES, Maria das Graças Volpe. Lematização versus Stemming. **USP, UFSCar, UNESP, São Carlos, São Paulo**, 2002.

DEERWESTER, Scott; DUMAIS, Susan T; FURNAS, George W; LANDAUER, Thomas K; HARSHMAN, Richard. Indexing by latent semantic analysis. **Journal of the American society for information science**, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990.

DENG, Li; LIU, Yang. **Deep learning in natural language processing**. [*S.l.*]: Springer, 2018.

ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg; XU, Xiaowei *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. *In*: 34. v. 96, p. 226–231.

FANG, Anjie. Analysing Political Events on Twitter: Topic Modelling and User Community Classification. **SIGIR Forum**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 1, p. 38–39, 2021.

GAO, Huiji; BARBIER, Geoffrey; GOOLSBY, Rebecca. Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. **IEEE Intelligent Systems**, v. 26, n. 3, p. 10–14, 2011.

GAO, Wang; PENG, Min; WANG, Hua; ZHANG, Yanchun; XIE, Qianqian; TIAN, Gang. Incorporating word embeddings into topic modeling of short text. **Knowledge and Information Systems**, Springer, v. 61, p. 1123–1145, 2019.

GHAFARI, Seyed Mohssen; TJORTJIS, Christos. A survey on association rules mining using heuristics. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 9, n. 4, 2019.

GREENE, Derek; CROSS, James P. Unveiling the political agenda of the european parliament plenary: A topical analysis. *In*: PROCEEDINGS of the ACM web science conference. [*S.l.*: *s.n.*], 2015. P. 1–10.

HADI, Mohammad Abdul; FARD, Fatemeh H. Aobtm: Adaptive online biterm topic modeling for version sensitive short-texts analysis. *In*: IEEE. 2020 IEEE international conference on software maintenance and evolution (ICSME). [*S.l.*: *s.n.*], 2020. P. 593–604.

HAGEN, Loni; UZUNER, Özlem; KOTFILA, Christopher; HARRISON, Teresa M; LAMANNA, Dan. Understanding citizens' direct policy suggestions to the federal government: A natural language processing and topic modeling approach. *In*: IEEE. 2015 48th Hawaii International Conference on System Sciences. [*S.l.*: *s.n.*], 2015. P. 2134–2143.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. Cluster Analysis: Basic Concepts and Methods. *In*: HAN, Jiawei; KAMBER, Micheline; PEI, Jian (Ed.). **Data Mining (Third Edition)**. Third Edition. Boston: Morgan Kaufmann, 2012. (The Morgan Kaufmann Series in Data Management Systems). P. 443–495.

HANSEN, Per Christian. The truncated SVD as a method for regularization. **BIT Numerical Mathematics**, Springer, v. 27, p. 534–553, 1987.

HARTIGAN, John A. **Clustering algorithms**. [*S.l.*]: John Wiley & Sons, Inc., 1975.

HAUPT, Michael Robert; JINICH-DIAMANT, Alex; LI, Jiawei; NALI, Matthew; MACKEY, Tim K. Characterizing twitter user topics and communication network dynamics of the “Liberate” movement during COVID-19 using unsupervised machine learning and social network analysis. **Online Social Networks and Media**, Elsevier, v. 21, p. 100–114, 2021.

HIMELBOIM, Itai; SMITH, Marc A; RAINIE, Lee; SHNEIDERMAN, Ben; ESPINA, Camila. Classifying Twitter topic-networks using social network analysis. **Social media+ society**, SAGE Publications Sage UK: London, England, v. 3, n. 1, 2017.

HIRAN, Kamal Kant; JAIN, Ritesh Kumar; LAKHWANI, Kamlesh; DOSHI, Ruchi. **Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)**. [*S.l.*]: BPB Publications, 2021.

HIRSCHBERG, Julia; MANNING, Christopher D. Advances in natural language processing. **Science**, American Association for the Advancement of Science, v. 349, n. 6245, p. 261–266, 2015.

HJORTH, L.; HINTON, S. **Understanding Social Media**. [*S.l.*]: SAGE Publications, 2019. ISBN 9781526426260.

HOFMANN, Thomas. Probabilistic latent semantic indexing. *In*: PROCEEDINGS of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. [*S.l.*: *s.n.*], 1999. P. 50–57.

HONG, Liangjie; DAVISON, Brian D. Empirical Study of Topic Modeling in Twitter. *In*: PROCEEDINGS of the First Workshop on Social Media Analytics. New York, NY, USA: Association for Computing Machinery, 2010. (SOMA ’10), p. 80–88.

HOSPEDALES, Timothy; GONG, Shaogang; XIANG, Tao. Video behaviour mining using a dynamic topic model. **International journal of computer vision**, Springer, v. 98, p. 303–323, 2012.

HOSSAIN, Tamanna; LOGAN IV, Robert L.; UGARTE, Arjuna; MATSUBARA, Yoshitomo; YOUNG, Sean; SINGH, Sameer. COVIDLies: Detecting COVID-19 Misinformation on Social Media. *In*: PROCEEDINGS of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020. [*S.l.*]: Association for Computational Linguistics, 2020.

HUANG, Jiajia; PENG, Min; LI, Pengwei; HU, Zhiwei; XU, Chao. Improving biterm topic model with word embeddings. **World Wide Web**, Springer, v. 23, n. 6, p. 3099–3124, 2020.

HUANG, Xuan; WU, Lei; YE, Yinsong. A review on dimensionality reduction techniques. **International Journal of Pattern Recognition and Artificial Intelligence**, World Scientific, v. 33, n. 10, 2019.

HUNG, Phan Duy; NGOC, Nguyen Duc; HANH, Tran Duc. K-means clustering using RA case study of market segmentation. *In*: PROCEEDINGS of the 2019 5th International Conference on E-Business and Applications. [*S.l.*: *s.n.*], 2019. P. 100–104.

INDURKHYA, Nitin; DAMERAU, Fred J. **Handbook of natural language processing**. [*S.l.*]: Chapman e Hall/CRC, 2010.

JIPKATE, Bharati R; GOHOKAR, VV. A comparative analysis of fuzzy c-means clustering and k means clustering algorithms. **International Journal Of Computational Engineering Research**, Citeseer, v. 2, n. 3, p. 737–739, 2012.

JU, Yiting; ADAMS, Benjamin; JANOWICZ, Krzysztof; HU, Yingjie; YAN, Bo; MCKENZIE, Grant. Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. *In*: SPRINGER. KNOWLEDGE Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20. [*S.l.*: *s.n.*], 2016. P. 353–367.

KARAMIZADEH, Sasan; ABDULLAH, Shahidan M; MANAF, Azizah A; ZAMANI, Mazdak; HOOMAN, Alireza. An overview of principal component analysis.

Journal of Signal and Information Processing, Scientific Research Publishing, v. 4, 3B, p. 173, 2013.

KAUFMAN, Leonard; ROUSSEEUW, Peter J. **Partitioning around medoids (program PAM)**. Hoboken, NJ, USA: John Wiley & Sons, Inc.: Wiley Series in Probability e Statistics, 1990. P. 68–125.

KHERWA, Pooja; BANSAL, Poonam. Topic Modeling: A Comprehensive Review. **EAI Endorsed Transactions on Scalable Information Systems**, EAI, v. 7, n. 24, jul. 2019.

LAFFERTY, John D; MCCALLUM, Andrew; PEREIRA, Fernando CN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *In*: PROCEEDINGS of the 18th International Conference on Machine Learning 2001 (ICML 2001). [*S.l.*: *s.n.*], 2001. P. 282–289.

LATIF, Jahanzaib; XIAO, Chuangbai; IMRAN, Azhar; TU, Shanshan. Medical imaging using machine learning and deep learning algorithms: a review. *In*: IEEE. 2019 2nd International conference on computing, mathematics and engineering technologies (iCoMET). [*S.l.*: *s.n.*], 2019. P. 1–5.

LAU, Jey Han; NEWMAN, David; BALDWIN, Timothy. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. *In*: PROCEEDINGS of the 14th Conference of the European Chapter of the Association for Computational Linguistics. Gothenburg, Sweden: Association for Computational Linguistics, 2014. P. 530–539.

LEDFORD, Heidi. **SOCIAL SCIENTISTS BATTLE BOTS TO GLEAN INSIGHTS ONLINE**. v. 578. [*S.l.*]: NATURE RESEARCH HEIDELBERGER PLATZ 3, BERLIN, 14197, GERMANY, 2020.

LEE, Daniel D; SEUNG, H Sebastian. Learning the parts of objects by non-negative matrix factorization. **Nature**, Nature Publishing Group UK London, v. 401, n. 6755, p. 788–791, 1999.

LEE, June-Goo; JUN, Sanghoon; CHO, Young-Won; LEE, Hyunna; KIM, Guk Bae; SEO, Joon Beom; KIM, Namkug. Deep learning in medical imaging: general overview. **Korean journal of radiology**, The Korean Society of Radiology, v. 18, n. 4, p. 570–584, 2017.

LI, Chenliang; DUAN, Yu; WANG, Haoran; ZHANG, Zhiqian; SUN, Aixin; MA, Zongyang. Enhancing topic modeling for short texts with auxiliary word embeddings. **ACM Transactions on Information Systems (TOIS)**, ACM New York, NY, USA, v. 36, n. 2, p. 1–30, 2017.

LI, Ximing; WANG, Yang; OUYANG, Jihong; WANG, Meng. Topic extraction from extremely short texts with variational manifold regularization. **Machine Learning**, Springer, v. 110, p. 1029–1066, 2021.

LI, Ximing; ZHANG, Jiaojiao; OUYANG, Jihong. Dirichlet Multinomial Mixture with Variational Manifold Regularization: Topic Modeling over Short Texts. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 33, n. 01, p. 7884–7891, 2019.

LI, Yingying; SHEN, Bo. Research on sentiment analysis of microblogging based on LSA and TF-IDF. *In*: IEEE. 2017 3rd ieee international conference on computer and communications (iccc). [*S.l.*: *s.n.*], 2017. P. 2584–2588.

LIKAS, Aristidis; VLASSIS, Nikos; VERBEEK, Jakob J. The global k-means clustering algorithm. **Pattern recognition**, Elsevier, v. 36, n. 2, p. 451–461, 2003.

LIU, Lin; TANG, Lin; DONG, Wen; YAO, Shaowen; ZHOU, Wei. An overview of topic modeling and its current applications in bioinformatics. **SpringerPlus**, SpringerOpen, v. 5, n. 1, p. 1–22, 2016.

LU, Heng-Yang; XIE, Lu-Yao; KANG, Ning; WANG, Chong-Jun; XIE, Jun-Yuan. Don't Forget the Quantifiable Relationship between Words: Using Recurrent Neural Network for Short Text Topic Discovery. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 31, n. 1, 2017.

MCLACHLAN, Geoffrey J; RATHNAYAKE, Suren. On the number of components in a Gaussian mixture model. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 4, n. 5, p. 341–355, 2014.

MCQUILLAN, Liz; MCAWEENEY, Erin; BARGAR, Alicia; RUCH, Alex. Cultural convergence: Insights into the behavior of misinformation networks on twitter. **arXiv preprint arXiv:2007.03443**, 2020.

MIKOLOV, Tomas; SUTSKEVER, Ilya; CHEN, Kai; CORRADO, Greg S; DEAN, Jeff. Distributed Representations of Words and Phrases and their Compositionality. *In*:

BURGES, C.J.; BOTTOU, L.; WELLING, M.; GHAHRAMANI, Z.; WEINBERGER, K.Q. (Ed.). **Advances in Neural Information Processing Systems**. [*S.l.*]: Curran Associates, Inc., 2013. v. 26.

MIMNO, David; WALLACH, Hanna; TALLEY, Edmund; LEENDERS, Miriam; MCCALLUM, Andrew. Optimizing semantic coherence in topic models. *In*: PROCEEDINGS of the 2011 conference on empirical methods in natural language processing. [*S.l.*: *s.n.*], 2011. P. 262–272.

MOHR, John W; BOGDANOV, Petko. **Introduction—Topic models: What they are and why they matter**. v. 41. [*S.l.*]: Elsevier, 2013. P. 545–569.

MOON, T.K. The expectation-maximization algorithm. **IEEE Signal Processing Magazine**, v. 13, n. 6, p. 47–60, 1996.

MOSER, Flavia; COLAK, Recep; RAFIEY, Arash; ESTER, Martin. Mining cohesive patterns from graphs with feature vectors. *In*: SIAM. PROCEEDINGS of the 2009 SIAM international conference on data mining. [*S.l.*: *s.n.*], 2009. P. 593–604.

MURTAGH, Fionn; CONTRERAS, Pedro. Algorithms for hierarchical clustering: an overview. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 2, n. 1, p. 86–97, 2012.

NADKARNI, Prakash M; OHNO-MACHADO, Lucila; CHAPMAN, Wendy W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 18, n. 5, p. 544–551, 2011.

NASSIF, Ali Bou; TALIB, Manar Abu; NASIR, Qassim; DAKALBAB, Fatima Mohamad. Machine learning for anomaly detection: A systematic review. **Ieee Access**, IEEE, v. 9, p. 78658–78700, 2021.

NATURE. **The powers and perils of using digital data to understand human behaviour**. [*S.l.*: *s.n.*], 2021. Disponível em: `https://www.nature.com/articles/d41586-021-01736-y/`. Acesso em: 16.02.2023.

NIGAM, Kamal; MCCALLUM, Andrew Kachites; THRUN, Sebastian; MITCHELL, Tom. Text classification from labeled and unlabeled documents using EM. **Machine learning**, Springer, v. 39, p. 103–134, 2000.

NIKITA, Murzintcev. Select number of topics for LDA model. **CRAN R Project**, 2016.

NILASHI, Mehrbakhsh; SAMAD, Sarminah; MINAEI-BIDGOLI, Behrouz; GHABBAN, Fahad; SUPRIYANTO, Eko. Online reviews analysis for customer segmentation through dimensionality reduction and deep learning techniques. **Arabian Journal for Science and Engineering**, Springer, v. 46, n. 9, p. 8697–8709, 2021.

NUGROHO, Robertus; PARIS, Cecile; NEPAL, Surya; YANG, Jian; ZHAO, Weiliang. A survey of recent methods on deriving topics from Twitter: algorithm to evaluation. **Knowledge and information systems**, Springer, v. 62, n. 7, p. 2485–2519, 2020.

O'CALLAGHAN, Derek; GREENE, Derek; CARTHY, Joe; CUNNINGHAM, Pádraig. An analysis of the coherence of descriptors in topic modeling. **Expert Systems with Applications**, v. 42, n. 13, p. 5645–5657, 2015.

ORDUN, Catherine; PURUSHOTHAM, Sanjay; RAFF, Edward. Exploratory analysis of covid-19 tweets using topic modeling, umap, and digraphs. **arXiv preprint arXiv:2005.03082**, 2020.

PAPASAVVA, Antonis; ZANNETTOU, Savvas; DE CRISTOFARO, Emiliano; STRINGHINI, Gianluca; BLACKBURN, Jeremy. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. *In*: PROCEEDINGS of the International AAAI Conference on Web and Social Media. [*S.l.*: *s.n.*], 2020. v. 14, p. 885–894.

PATEL, Ankur A. **Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data**. [*S.l.*]: O'Reilly Media, 2019.

PATHAK, Ajeet Ram; PANDEY, Manjusha; RAUTARAY, Siddharth. Adaptive model for dynamic and temporal topic modeling from big data using deep learning architecture. **International Journal of Intelligent Systems and Applications**, Modern Education e Computer Science Press, v. 9, n. 6, p. 13, 2019.

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine learning in Python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

POPESCU, Marius-Constantin; BALAS, Valentina E; PERESCU-POPESCU, Liliana; MASTORAKIS, Nikos. Multilayer perceptron and neural networks. **WSEAS**

Transactions on Circuits and Systems, World Scientific, Engineering Academy e Society (WSEAS), v. 8, n. 7, p. 579–588, 2009.

QIANG, Jipeng; QIAN, Zhenyu; LI, Yun; YUAN, Yunhao; WU, Xindong. Short text topic modeling techniques, applications, and performance: a survey. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 34, n. 3, p. 1427–1445, 2020.

QUAN, Xiaojun; KIT, Chunyu; GE, Yong; PAN, Sinno Jialin. Short and sparse text topic modeling via self-aggregation. *In*: AAAI PRESS/INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE. 24TH International Joint Conference on Artificial Intelligence, IJCAI 2015. [*S.l.*: *s.n.*], 2015. P. 2270–2276.

RAJPUT, Adil. Natural language processing, sentiment analysis, and clinical analytics. *In*: INNOVATION in health informatics. [*S.l.*]: Elsevier, 2020. P. 79–97.

RANA, Toqir A; CHEAH, Yu-N; LETCHMUNAN, Sukumar. Topic Modeling in Sentiment Analysis: A Systematic Review. **Journal of ICT Research & Applications**, v. 10, n. 1, 2016.

RASHID, Junaid; SHAH, Syed Muhammad Adnan; IRTAZA, Aun. Fuzzy topic modeling approach for text mining over short text. **Information Processing & Management**, Elsevier, v. 56, n. 6, 2019.

REHOREK, Radim. **Gensim topic modeling for humans: Core Concepts**. [*S.l.*: *s.n.*], 2022. Disponível em: `https://radimrehurek.com/gensim/auto_examples/core/run_core_concepts.html`. Acesso em: 13 maio 2023.

ŘEHŮŘEK, Radim; SOJKA, Petr. Software Framework for Topic Modelling with Large Corpora. *In*: PROCEEDINGS of LREC 2010 workshop New Challenges for NLP Frameworks. Valletta, Malta: University of Malta, 2010. P. 46–50.

ROSNER, Frank; HINNEBURG, Alexander; RÖDER, Michael; NETTLING, Martin; BOTH, Andreas. Evaluating topic coherence measures. **CoRR**, abs/1403.6397, 2014.

RÜDIGER, Matthias; ANTONS, David; JOSHI, Amol M.; SALGE, Torsten-Oliver. Topic modeling revisited: New evidence on algorithm performance and quality metrics. **PLOS ONE**, Public Library of Science, v. 17, n. 4, p. 1–25, 2022.

SALTON, G. Some research problems in automatic information retrieval. *In*: PROCEEDINGS of the 6th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '83. [*S.l.*: *s.n.*], 1983.

SANTAELLA, L; LEMOS, R. Redes sociais digitais: a cognição conectiva do Twitter, 2010.

SANTRA, S.; BHOWMICK, S.; PAUL, A.; CHATTERJEE, P.; DEYASI, A. Development of GUI for text-to-speech recognition using natural language processing. *In*: IEEE. 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech). [*S.l.*: *s.n.*], 2018. P. 1–4.

SCHMARJE, Lars; SANTAROSSA, Monty; SCHRÖDER, Simon-Martin; KOCH, Reinhard. A survey on semi-, self-and unsupervised learning for image classification. **IEEE Access**, IEEE, v. 9, p. 82146–82168, 2021.

SHAHNAZ, Farial; BERRY, Michael W.; PAUCA, V.Paul; PLEMMONS, Robert J. Document clustering using nonnegative matrix factorization. **Information Processing Management**, v. 42, n. 2, p. 373–386, 2006.

SIDDIQUI, Z; RATHINAM, F. **Big data in the time of a pandemic**. [*S.l.*]: News e Press Release, 2021. Disponível em: `https://www.3ieimpact.org/blogs/big-data-time-pandemic`.

SIMPSON, Sean S; ADAMS, Nikki; BRUGMAN, Claudia M; CONNERS, Thomas J. Detecting novel and emerging drug terms using natural language processing: a social media corpus study. **JMIR public health and surveillance**, JMIR Publications Inc., Toronto, Canada, v. 4, n. 1, 2018.

SIMULA, Henri; TÖLLINEN, Aarne; KARJALUOTO, Heikki. Crowdsourcing in the social media era: A case study of industrial marketers. **Journal of Marketing Development and Competitiveness**, North American Business Press, v. 7, n. 2, 2013.

STATISTA. **Number of social network users worldwide from 2017 to 2025**. [*S.l.*: *s.n.*], 2020. Business Data Platform. Disponível em: `https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/`. Acesso em: 06.02.2023.

STEYVERS, Mark; GRIFFITHS, Tom. Probabilistic topic models. *In*: HANDBOOK of latent semantic analysis. [*S.l.*]: Psychology Press, 2007. P. 439–460.

SUN, Xuan; JIANG, Longquan; ZHANG, Minghuan; WANG, Cheng; CHEN, Ying. Unsupervised Learning for Product Ontology from Textual Reviews on E-Commerce Sites. *In*: PROCEEDINGS of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence. [*S.l.*: *s.n.*], 2019. P. 260–264.

SZABÓ, Pèter; GENGE, Béla. Efficient conversion prediction in E-Commerce applications with unsupervised learning. *In*: IEEE. 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). [*S.l.*: *s.n.*], 2020. P. 1–6.

TIXIER, Antoine J-P; HALLOWELL, Matthew R; RAJAGOPALAN, Balaji; BOWMAN, Dean. Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports. **Automation in Construction**, Elsevier, v. 62, p. 45–56, 2016.

TURING, A. M. Computing Machinery and Intelligence. **Mind**, v. LIX, n. 236, p. 433–460, 1950.

TUTEN, T.L. **Social Media Marketing**. [*S.l.*]: SAGE Publications, 2020. (Core textbook). ISBN 9781529736229.

VAYANSKY, Ike; KUMAR, Sathish A.P. A review of topic modeling methods. **Information Systems**, v. 94, p. 101582, 2020.

VORONTSOV, Konstantin; POTAPENKO, Anna. Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization. *In*: IGNATOV, Dmitry I.; KHACHAY, Mikhail Yu.; PANCHENKO, Alexander; KONSTANTINOVA, Natalia; YAVORSKY, Rostislav E. (Ed.). **Analysis of Images, Social Networks and Texts**. [*S.l.*]: Springer International Publishing, 2014. P. 29–46.

WALLACH, Hanna M.; MURRAY, Iain; SALAKHUTDINOV, Ruslan; MIMNO, David. Evaluation Methods for Topic Models. *In*: PROCEEDINGS of the 26th Annual International Conference on Machine Learning. Montreal, Quebec, Canada: ACM, 2009. (ICML '09), p. 1105–1112.

WAN, Xing. Application of K-means Algorithm in Image Compression. *In*: IOP PUBLISHING, 5. IOP Conference Series: Materials Science and Engineering. [*S.l.*: *s.n.*], 2019. v. 563.

WILLIAMS, Hywel TP; MCMURRAY, James R; KURZ, Tim; LAMBERT, F Hugo. Network analysis reveals open forums and echo chambers in social media discussions of climate change. **Global environmental change**, Elsevier, v. 32, p. 126–138, 2015.

WU, Guohua; LIN, Hairong; FU, Ershuai; WANG, Liuyang. An improved k-means algorithm for document clustering. *In*: IEEE. 2015 international conference on computer science and mechanical automation (CSMA). [*S.l.*: *s.n.*], 2015. P. 65–69.

WU, Pei-Ju; LIN, Kun-Chen. Unstructured big data analytics for retrieving e-commerce logistics knowledge. **Telematics and Informatics**, Elsevier, v. 35, n. 1, p. 237–244, 2018.

WU, Xiaobao; LI, Chunping. Short text topic modeling with flexible word patterns. *In*: IEEE. 2019 International Joint Conference on Neural Networks (IJCNN). [*S.l.*: *s.n.*], 2019. P. 1–7.

WU, Xiaobao; LI, Chunping; ZHU, Yan; MIAO, Yishu. Short Text Topic Modeling with Topic Distribution Quantization and Negative Sampling Decoder. *In*: PROCEEDINGS of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). [*S.l.*]: Association for Computational Linguistics, 2020. P. 1772–1782.

XU, Dongkuan; TIAN, Yingjie. A comprehensive survey of clustering algorithms. **Annals of Data Science**, Springer, v. 2, p. 165–193, 2015.

YADAV, Mayank; JOSHI, Yatish; RAHMAN, Zillur. Mobile Social Media: The New Hybrid Element of Digital Marketing Communications. **Procedia - Social and Behavioral Sciences**, v. 189, p. 335–343, 2015. Operations Management in Digital Economy. ISSN 1877-0428.

YIN, Jianhua; WANG, Jianyong. A dirichlet multinomial mixture model-based approach for short text clustering. *In*: PROCEEDINGS of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. [*S.l.*: *s.n.*], 2014. P. 233–242.

ZHANG, Yin; CHEN, Min; HUANG, Dijiang; WU, Di; LI, Yong. iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. **Future Generation Computer Systems**, Elsevier, v. 66, p. 30–35, 2017.

ZHANG, Yin; JIN, Rong; ZHOU, Zhi-Hua. Understanding bag-of-words model: a statistical framework. **International journal of machine learning and cybernetics**, Springer, v. 1, p. 43–52, 2010.

ZHAO, He; DU, Lan; BUNTINE, Wray; LIU, Gang. MetaLDA: A Topic Model that Efficiently Incorporates Meta Information. *In*: 2017 IEEE International Conference on Data Mining (ICDM). [*S.l.*: *s.n.*], 2017. P. 635–644.

ZHAO, He; DU, Lan; LIU, Guanfeng; BUNTINE, Wray. Leveraging meta information in short text aggregation. *In*: PROCEEDINGS of the 57th Annual Meeting of the Association for Computational Linguistics. [*S.l.*: *s.n.*], 2019. P. 4042–4049.

ZHAO, Wayne Xin; JIANG, Jing; WENG, Jianshu; HE, Jing; LIM, Ee-Peng; YAN, Hongfei; LI, Xiaoming. Comparing Twitter and Traditional Media Using Topic Models. *In*: CLOUGH, Paul; FOLEY, Colum; GURRIN, Cathal; JONES, Gareth J. F.; KRAAIJ, Wessel; LEE, Hyowon; MUDOCH, Vanessa (Ed.). **Advances in Information Retrieval**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. P. 338–349.

ZHENG, Xin; LEI, Qinyi; YAO, Run; GONG, Yifei; YIN, Qian. Image segmentation based on adaptive K-means algorithm. **EURASIP Journal on Image and Video Processing**, Springer, v. 2018, n. 1, p. 1–10, 2018.

ZHOU, Zhengzhong; ZHOU, Jingjin; ZHANG, Liqing. Demand-adaptive clothing image retrieval using hybrid topic model. *In*: PROCEEDINGS of the 24th ACM international conference on Multimedia. [*S.l.*: *s.n.*], 2016. P. 496–500.

ZONG, Zhaorong; HONG, Changchun. On application of natural language processing in machine translation. *In*: IEEE. 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE). [*S.l.*: *s.n.*], 2018. P. 506–510.

ZUO, Yuan; WU, Junjie; ZHANG, Hui; LIN, Hao; WANG, Fei; XU, Ke; XIONG, Hui. Topic Modeling of Short Texts: A Pseudo-Document View. *In*: PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 2016. P. 2105–2114.

APÊNDICE A – RELATÓRIO DOS RESULTADOS OBTIDOS NA ANÁLISE DOS ALGORITMOS DE MODELAGEM DE TÓPICOS USANDO TEXTOS DE MÍDIAS SOCIAIS.

Este relatório apresenta os desempenhos dos 16 algoritmos de modelagem de tópicos selecionados da literatura ao serem avaliados por 5 métricas distintas. Os algoritmos foram divididos entre quatro categorias, sendo três delas identificadas por Qiang *et al.* (2020). Além disso, na literatura constantemente adota-se outro grupo de algoritmos denominados como “Tradicionais”. Portanto, com base nisso, a divisão dos algoritmos se dá pelas seguintes categorias:

- Tradicionais (LDA, LSA, PLSA, NMF);
- Baseados em DMM (DMM, GPU-PDMM, NQTM, LapDMM);
- Baseados em auto agregação (SATM, CRFTM, PTM, MIGA);
- Baseados em co-ocorrência global (BTM, WNTM, MTM, NBTMWE).

Cada processo foi avaliado considerando as métricas de avaliação de perplexidade, distância de tópicos, divergência-KL simétrica, e coerência de tópicos (C_{UMASS} e C_{W2V}).

A perplexidade mede o quão bem um modelo consegue desempenhar num conjunto de testes. Valores menores de perplexidade indicam uma melhor capacidade de generalização do modelo, enquanto valores maiores indicam *overfitting*.

A distância de tópicos mede a coesão entre os tópicos gerados. Valores maiores indicam que as distribuições possuem mais diferenças, e valores menores indicam igualdades entre elas. Essa é uma forma de identificar modelos que geram tópicos sobrepostos, quando os tópicos compartilham palavras e termos semelhantes e, consequentemente, abordam assuntos e temas semelhantes.

A divergência-KL simétrica mede a diferença entre os tópicos, semelhante à distância de tópicos. Porém, a divergência-KL é mais sensível a pequenas diferenças entre as distribuições. Valores menores indicam maior similaridade entre os tópicos, e valores maiores indicam maior diferença entre os tópicos. Nota-se que, a distância é particularmente útil para comparar a direção de vetores de tópicos, enquanto a divergência-KL simétrica é adequada para comparar distribuições de probabilidade de tópicos. Usar ambas as métricas pode fornecer uma visão mais completa do desempenho de uma modelagem de tópicos e ajudar a identificar tópicos distintos e interpretáveis. Ademais, valores maiores da divergência-KL indicam maior dissimilaridade entre os tópicos.

A coerência dos tópicos é medida de duas formas que visam avaliar a interpretabilidade e relevância dos tópicos gerados:

- C_{UMASS}: com base na razão entre as co-ocorrências de pares de palavras

nos tópicos e as contagens individuais das palavras; e

- C_{W2V}: com base na semelhança semântica média entre as palavras em um tópico usando vetores de palavras gerados por um modelo Word2Vec. A semelhança é calculada com a distância de cossenos.

Nos dois casos de coerência, tópicos com resultados maiores são mais interpretáveis e relevantes, por conterem palavras fortemente relacionadas no contexto dos documentos (C_{UMASS}), e palavras que estão relacionadas semanticamente (C_{W2V}).

A coerência C_{W2V} é especialmente útil quando se considera a semântica das palavras e não apenas a frequência de co-ocorrência nos documentos. Essa abordagem fornece uma visão mais rica da qualidade e interpretabilidade dos tópicos gerados, especialmente em casos onde a co-ocorrência sozinha pode não ser suficiente para avaliar a relação entre as palavras.

À vista disso, as métricas supracitadas foram selecionadas para avaliar o desempenho de 16 algoritmos de modelagem de tópicos. Um corpus de textos foi produzido exclusivamente para esse trabalho. Trata-se de um dataset com aproximadamente 4 milhões de tuítes com idioma previamente classificado como português. Esse corpus foi produzido a partir do trabalho de Banda *et al.* (2021), e compõe-se de tuítes relacionados à COVID-19 publicados no período entre o dia 1 de março de 2020 até o dia 1 de março de 2021.

A.1 TRADICIONAIS (LDA, LSA, PLSA, NMF)

Os algoritmos tradicionais LDA, LSA, PLSA e NMF são amplamente utilizados para modelagem de tópicos em textos longos, como artigos científicos, livros ou relatórios. No entanto, quando aplicados a textos curtos, como tuítes ou mensagens de texto, esses algoritmos podem ter desempenho limitado devido à falta de contexto e coocorrência de palavras.

Dentre eles, o LDA é um modelo de tópicos frequentemente usado para analisar grandes conjuntos de documentos e encontrar os tópicos subjacentes nesses documentos. O LDA é um modelo probabilístico que assume que cada documento em um corpus é uma mistura de vários tópicos, e cada tópico é uma distribuição de palavras. Ou seja, o LDA assume que cada palavra em um documento é gerada por um dos tópicos presentes no documento, e que a distribuição desses tópicos é determinada pela distribuição de tópicos no corpus todo.

O LSA é uma técnica de processamento de linguagem natural que permite identificar tópicos relevantes em um conjunto de documentos. Ele funciona a partir da representação vetorial dos documentos em um espaço n-dimensional, no qual cada termo é representado como um vetor. A técnica utiliza a análise de valores singulares para reduzir a dimensionalidade da matriz de documentos-termos e identificar os

tópicos mais relevantes. A partir da análise dos vetores de termos, o LSA permite a modelagem de tópicos e a interpretação dos resultados obtidos. Além disso, ele considera o peso dos termos e utiliza técnicas como o TF-IDF para atribuir pesos aos termos no cálculo da similaridade entre os documentos.

O PLSA é um modelo probabilístico para análise de tópicos em textos. Ele pode identificar os tópicos latentes que compõem um conjunto de documentos, bem como a distribuição de probabilidade das palavras em cada tópico e a distribuição de probabilidade dos tópicos em cada documento. A abordagem básica do PLSA consiste em modelar a geração de um documento por meio de uma distribuição de probabilidade condicional, representada por uma mistura de distribuições de probabilidade de tópicos. Cada tópico, por sua vez, é modelado por meio de uma distribuição de probabilidade condicional de palavras. Dessa forma, o PLSA considera que cada documento é gerado pela seleção de um tópico e a seleção de palavras condicionais a esse tópico.

O NMF é uma técnica de fatorização de matrizes que tem sido amplamente utilizada em problemas de mineração de dados e análise de dados textuais. O objetivo do NMF é encontrar duas matrizes não negativas, uma matriz W de dimensão $m \times r$, e uma matriz H de dimensão $r \times n$, que aproximem a matriz original X de dimensão $m \times n$, tal que $X \approx WH$. O NMF é particularmente útil em problemas de redução de dimensionalidade, onde é necessário encontrar representações mais compactas de dados, e em problemas de agrupamento e classificação, onde a interpretabilidade dos componentes é importante. A restrição de não-negatividade das matrizes W e H garante a interpretabilidade dos componentes. No entanto, a dependência da inicialização dos fatores é uma desvantagem do NMF, e o resultado pode variar dependendo da escolha dos parâmetros e da inicialização dos fatores.

Assim sendo, o grupo de algoritmos tradicionais foi analisado usando as 5 métricas de avaliação. A seguir, os resultados de cada análise são demonstrados e discorridos. Primeiramente será exibido um gráfico que ilustra o desempenho de cada algoritmo da categoria nos testes, e depois uma tabela com todos os resultados.

A.1.1 Perplexidade

A análise da perplexidade dos algoritmos tradicionais revela que, dentre eles, o LDA apresenta a melhor capacidade de generalização, seguido pelo LSA, PLSA, e NMF. A Figura 22 ilustra o desempenho de perplexidade de cada algoritmo, sendo K o número de tópicos.

A Tabela 5 mostra em detalhes os valores da perplexidade de cada algoritmo. Para melhor visualização dos dados, os valores foram arredondados para três casas decimais. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

Observa-se que a perplexidade de todos os algoritmos Tradicionais aumenta à

Figura 22 – Perplexidade dos algoritmos Tradicionais.

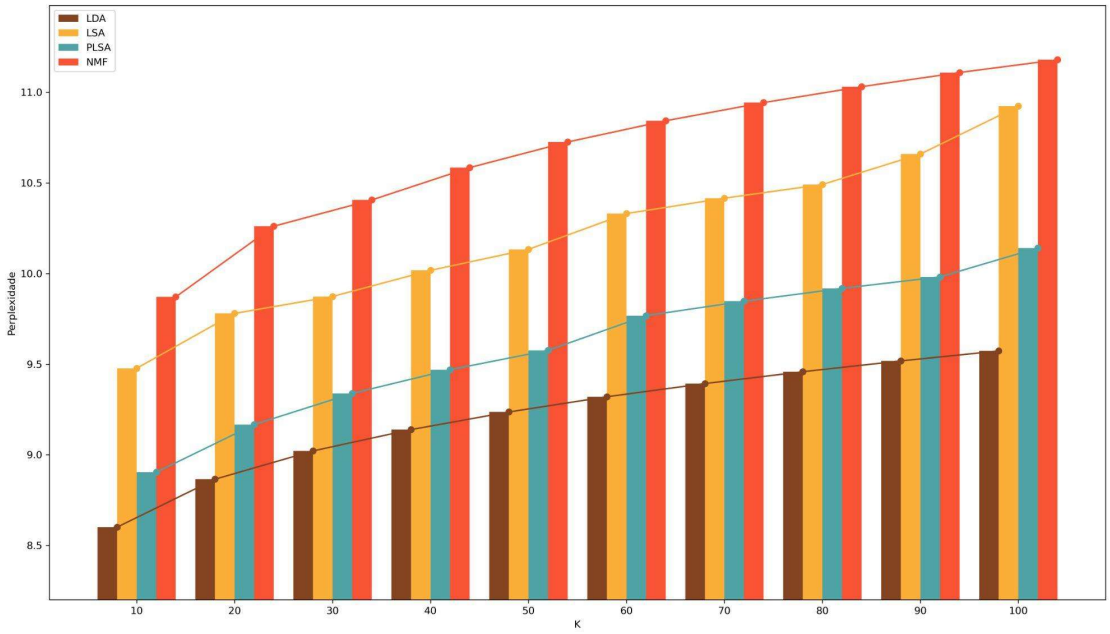


Tabela 5 – Perplexidade dos algoritmos Tradicionais para cada valor de *K*.

Algoritmo	**Perplexidade para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
LDA	8.6	8.865	9.021	9.139	9.236	9.32	9.393	9.458	9.518	**9.573**
LSA	9.477	9.780	9.873	10.017	10.133	10.331	10.416	10.491	10.659	**10.923**
PLSA	8.903	9.167	9.338	9.469	9.576	9.767	9.847	9.918	9.981	**10.141**
NMF	9.871	10.261	10.506	10.684	10.826	10.943	11.043	11.131	11.209	**11.279**

medida que *K* aumenta. Este aumento na perplexidade é esperado, já que a incerteza na atribuição de palavras aos tópicos tende a aumentar com o maior número de tópicos.

Entre os algoritmos tradicionais, o NMF apresenta a maior perplexidade em todos os números de tópicos, indicando a dificuldade que este algoritmo possui para gerar um modelo com boa generalização no contexto de textos curtos. O LSA e PLSA tem um desempenho intermediário enquanto o LDA apresentou a melhor performance em geral e escalabilidade nesse quesito.

A.1.2 Distância de tópicos

A análise de distância de tópicos dos algoritmos tradicionais mostra que os algoritmos LSA, PLSA e NMF tendem a gerar tópicos mais distintos e menos sobrepostos em comparação com o LDA. A Figura 23 ilustra o desempenho de distância de tópicos de cada algoritmo, sendo *K* o número de tópicos.

A Tabela 6 mostra os valores de distância de tópicos para os algoritmos Tradicionais. Os maiores valores de cada algoritmo foram destacados com negrito, e os

Figura 23 – Distância de tópicos dos algoritmos Tradicionais.

menores valores foram sublinhados.

Tabela 6 – Distância de tópicos dos algoritmos Tradicionais para cada valor de K.

Algoritmo	**Distância de tópicos para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
LDA	**0.167**	0.116	0.124	0.068	0.065	0.062	0.085	0.081	0.094	0.053
LSA	**0.139**	0.125	0.133	0.066	0.1	0.099	0.104	0.106	0.095	0.071
PLSA	**0.14**	0.116	0.117	0.065	0.081	0.081	0.071	0.114	0.05	0.096
NMF	**0.141**	0.134	0.135	**0.141**	0.089	0.104	0.098	0.098	0.092	0.075

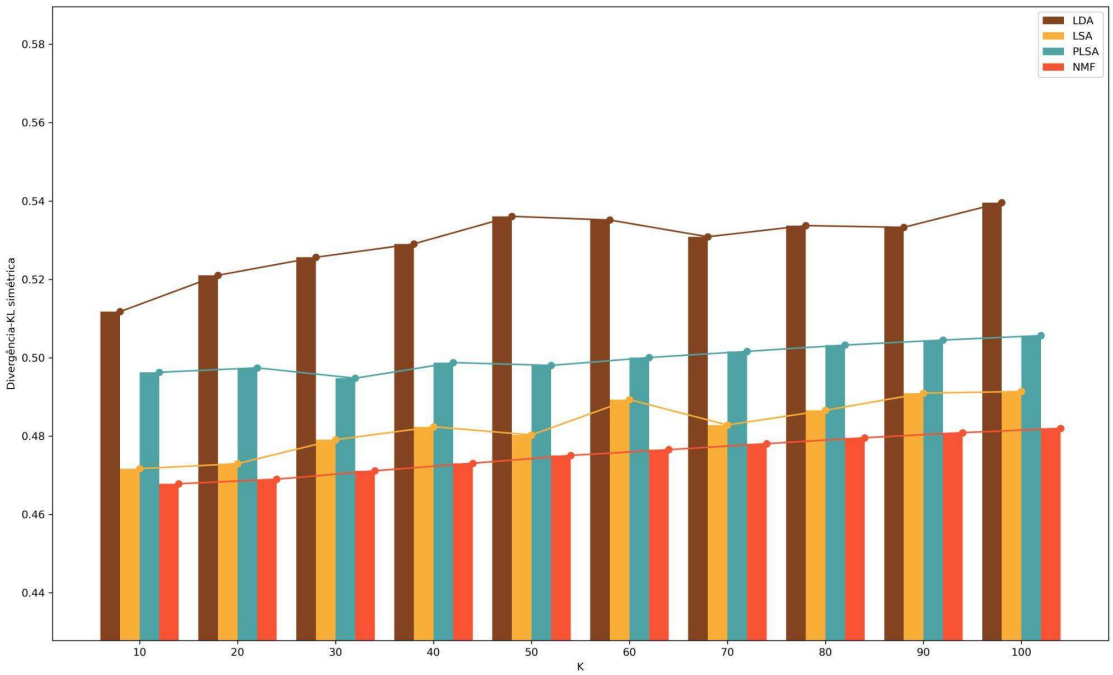
Em linhas gerais, os resultados de todos os algoritmos diminuíram com o aumento de K. O desempenho do LDA é mais suscetível ao valor de K, tendo a maior variação de distância de tópicos entre os Tradicionais, apesar de, em alguns casos, gerar os tópicos mais sobrepostos. O PLSA apresenta valores semelhantes aos do LSA, com uma ligeira vantagem em gerar tópicos mais distintos em alguns casos (por exemplo, com 90 tópicos). Já o NMF apresenta valores semelhantes aos do LSA e PLSA, mas com uma tendência a gerar tópicos ligeiramente mais sobrepostos.

A.1.3 Divergência-KL simétrica

A análise da divergência-KL simétrica mostra que o LSA e o NMF parecem gerar distribuições de tópicos mais similares em comparação com o LDA e o PLSA. A Figura 24 mostra o desempenho da divergência-KL simétrica de cada algoritmo, sendo K o

número de tópicos. A Figura 24 mostra os resultados de divergência dos algoritmos Tradicionais.

Figura 24 – Divergência-KL simétrica dos algoritmos Tradicionais.



A Tabela 7 mostra os valores da divergência-KL simétrica para os algoritmos tradicionais. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

Tabela 7 – Divergência-KL simétrica dos algoritmos Tradicionais para cada valor de K.

Algoritmo	**Divergência-KL simétrica para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
LDA	0.511	0.521	0.525	0.529	0.536	0.535	0.53	0.533	0.533	**0.539**
LSA	0.471	0.472	0.479	0.482	0.48	0.489	0.482	0.486	**0.491**	**0.491**
PLSA	0.496	0.497	0.495	0.499	0.498	0.500	0.502	0.503	0.504	**0.506**
NMF	0.468	0.469	0.471	0.473	0.475	0.476	0.478	0.479	0.481	**0.482**

A dissimilaridade entre as distribuições de tópicos geradas no LDA aumenta à medida que K aumenta. Para o LSA, os valores são consistentemente menores em comparação com o LDA, o que pode indicar que as distribuições de tópicos geradas pelo LSA são mais similares entre si. O PLSA tem valores intermediários em relação aos outros, gerando tópicos com dissimilaridade moderada. O NMF apresenta valores menores que os outros algoritmos, gerando tópicos mais similares entre si.

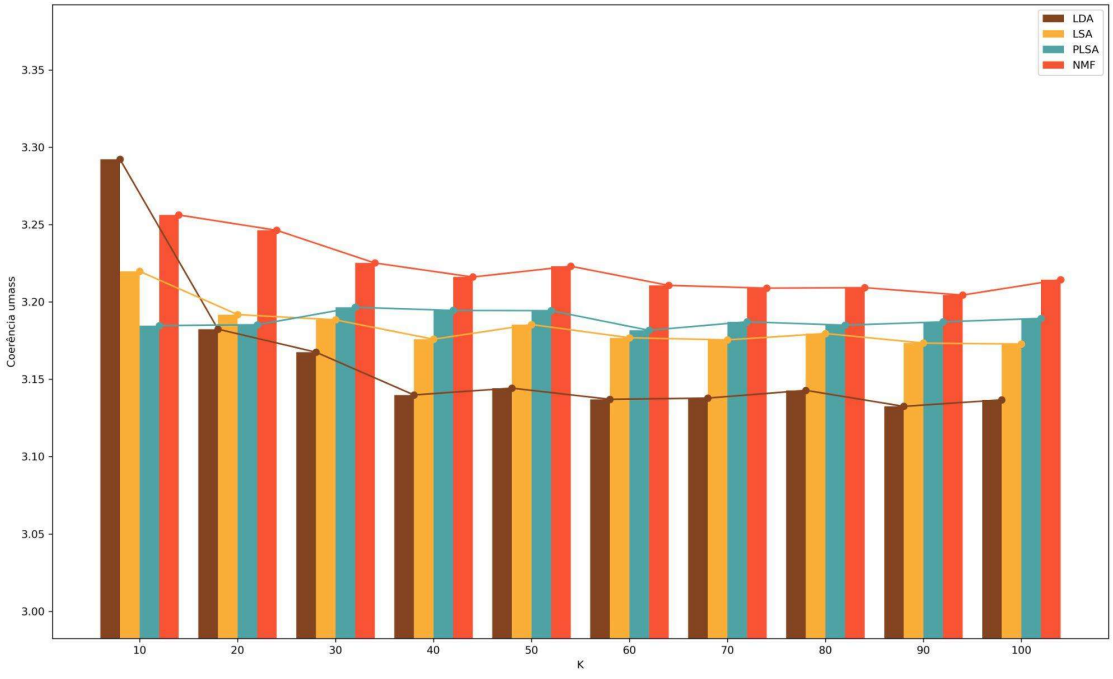
Assim, observa-se que o LDA tem resultados peculiares em termos de distância de tópicos e divergência-KL simétrica. Em geral, a distância de tópicos diminui à

medida que K aumenta, assim como a divergência-KL simétrica apresenta valores um pouco mais altos em comparação com outros algoritmos, sugerindo que as distribuições de probabilidade dos tópicos são mais distintas. Apesar disso, os tópicos gerados são sobrepostos. O PLSA apresenta uma distância de tópicos com tendência decrescente à medida que K aumenta. A divergência do PLSA é a mais próxima do LDA, mas apesar disso, os tópicos dele são menos distintos que os do LSA em alguns casos. Já o NMF mostra distância de tópicos semelhantes aos outros algoritmos em vários casos, apesar de consistentemente ter o resultado mais baixo de divergência.

A.1.4 Coerência C_{UMASS}

A análise dos valores de coerência C_{UMASS} mostra que o NMF gera tópicos com maior coerência em comparação com os outros. A Figura 25 ilustra a coerência C_{UMASS} dos algoritmos Tradicionais.

Figura 25 – Coerência C_{UMASS} dos algoritmos Tradicionais.



A Tabela 8 mostra os valores da coerência C_{UMASS} para os algoritmos Tradicionais. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

O LSA possui variação relativamente pequena na qualidade dos tópicos gerados com diferentes números de tópicos, enquanto o PLSA apresenta a menor variação na coerência entre os algoritmos. Dessa forma, esses algoritmos mantêm a consistência da coerência independente do valor de K. Além disso, os valores são consistentemente mais baixos em comparação com o NMF. Por fim, observa-se que os valores do NMF

Tabela 8 – Coerência C_{UMASS} dos algoritmos Tradicionais.

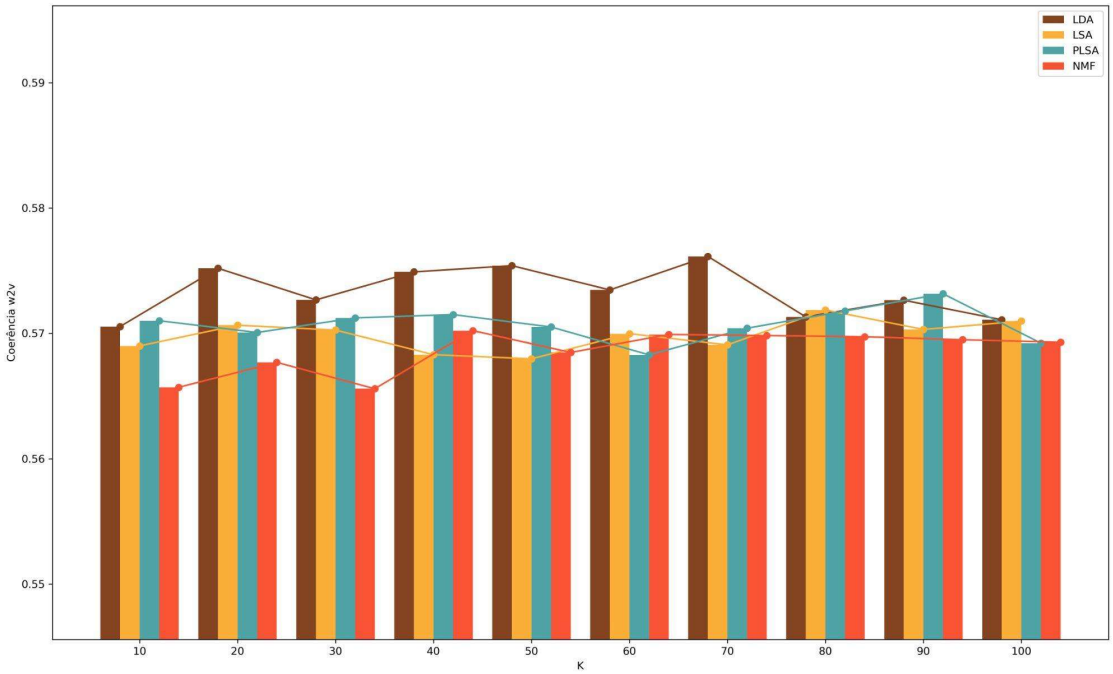
Algoritmo	**Coerência C_{UMASS} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
LDA	**3.292**	3.182	3.168	3.14	3.144	3.137	3.138	3.143	3.132	3.137
LSA	**3.22**	3.192	3.188	3.176	3.185	3.177	3.175	3.179	3.173	3.173
PLSA	3.185	3.185	**3.196**	3.194	3.194	3.181	3.187	3.185	3.187	3.189
NMF	**3.256**	3.246	3.225	3.216	3.223	3.211	3.209	3.209	3.204	3.214

são, frequentemente, maiores do que os do PLSA, mas menores do que os do LDA em uma ocasião (K = 10), sugerindo que a qualidade dos tópicos gerados pelo NMF é a mais consistente entre os algoritmos.

A.1.5 Coerência C_{W2V}

A análise dos valores de coerência C_{W2V} mostra que a qualidade dos tópicos gerados pelos algoritmos LDA, LSA e PLSA é semelhante, com o NMF apresentando valores ligeiramente mais baixos, e o LDA com valores ligeiramente mais altos. A Figura 26 mostra os resultados da análise de coerência C_{W2V} nos algoritmos Tradicionais.

Figura 26 – Coerência C_{W2V} dos algoritmos Tradicionais.



A Tabela 9 mostra os valores da coerência C_{W2V} para os algoritmos Tradicionais. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

Tabela 9 – Coerência C_{W2V} dos algoritmos Tradicionais.

Algoritmo	**Coerência C_{W2V} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
LDA	0.57	0.575	0.573	0.575	0.575	0.573	**0.576**	0.571	0.573	0.571
LSA	0.569	0.571	0.57	0.568	0.568	0.57	0.569	**0.572**	0.57	0.571
PLSA	0.571	0.57	0.571	0.572	0.57	0.568	0.57	0.572	**0.573**	0.569
NMF	0.566	0.568	0.566	**0.57**	0.568	**0.57**	0.567	**0.57**	0.569	0.569

Embora os valores da coerência C_{W2V} estejam próximos entre os algoritmos, ainda é possível identificar diferenças sutis no desempenho de cada algoritmo. Por exemplo, o LDA e o PLSA têm valores de coerência ligeiramente superiores em comparação com o LSA e o NMF. Todos os algoritmos tradicionais apresentaram valores de coerência C_{W2V} bastante próximos entre si. Portanto, em geral, eles geram tópicos com qualidade comparável em termos da similaridade semântica das palavras dentro de cada tópico.

Para todos os algoritmos, os valores da coerência C_{W2V} não mostram uma tendência clara de aumento ou diminuição à medida que o número de tópicos aumenta. Isso indica que cada algoritmo pode manter a coerência dos tópicos gerados em diferentes configurações de número de tópicos, embora possa haver variações pontuais na qualidade.

A.2 BASEADOS EM DMM (DMM, GPU-PDMM, NQTM, LAPDMM)

Os algoritmos baseados em DMM são frequentemente usados em contextos onde a análise de tópicos precisa ser realizada em coleções de documentos curtos ou onde a coocorrência de palavras é especialmente significativa para a identificação de padrões semânticos. Esses algoritmos se destacam na captura de relações semânticas mais sutis e na geração de tópicos mais coerentes e informativos em comparação com modelos de tópicos tradicionais.

O DMM é um modelo de mistura simples que assume que cada documento é gerado por um único tópico e cada palavra em um documento é gerada a partir da distribuição multinomial de palavras correspondentes a esse tópico. Este modelo é mais adequado para documentos curtos por considerar apenas as frequências de palavras nos documentos e não a coocorrência de palavras.

O GPU-PDMM é uma extensão do modelo DMM que incorpora a abordagem da *Generalized Polya urn* para melhorar a capacidade de modelar tópicos em documentos curtos. O GPU-PDMM assume que a distribuição de palavras em cada tópico segue uma distribuição de Poisson, permitindo uma maior flexibilidade na modelagem da frequência de palavras. Isso resulta em uma melhor representação dos tópicos, especialmente quando se lida com documentos curtos, onde a frequência de palavras

é geralmente baixa.

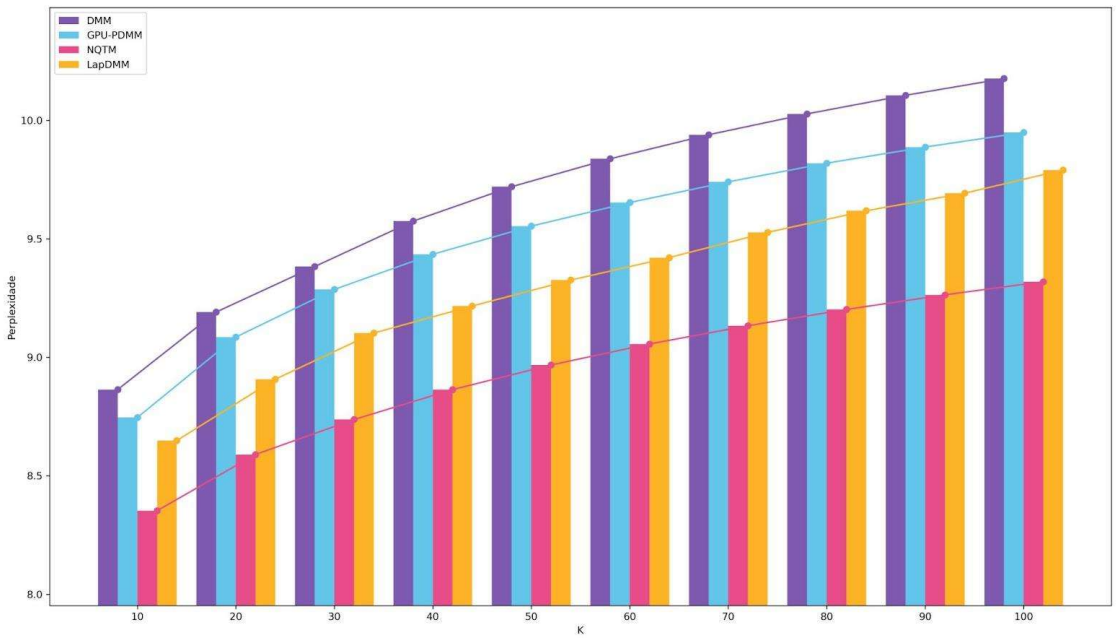
O NQTM é um modelo baseado em DMM que aplica duas técnicas para melhorar o desempenho na análise de tópicos em documentos curtos: amostragem negativa e quantização. A amostragem negativa permite que o modelo foque em palavras mais discriminativas durante o treinamento, ao passo que a quantização reduz a complexidade computacional e a quantidade de memória necessária para armazenar e processar o modelo.

O LapDMM é um modelo que combina o DMM com uma abordagem de regularização baseada em grafos. Ele utiliza a estrutura de grafo laplaciano para incorporar informações de proximidade entre os documentos, considerando as relações semânticas e estruturais entre os documentos e os tópicos. Assim sendo, o grupo de algoritmos baseados em DMM foi analisado usando as 5 métricas de avaliação. Os resultados de cada análise são demonstrados e discorridos a seguir. Um gráfico que ilustra o desempenho de cada é mostrado, e depois uma tabela com todos os resultados.

A.2.1 Perplexidade

Os testes mostram o NQTM apresenta a melhor capacidade de generalização, seguido pelo LapDMM, GPU-PDMM e, finalmente, DMM. A Figura 27 ilustra o desempenho de perplexidade de cada algoritmo, sendo K o número de tópicos.

Figura 27 – Perplexidade dos algoritmos baseados em DMM.



A Tabela 10 mostra em detalhes os valores da perplexidade de cada algoritmo. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

Tabela 10 – Perplexidade dos algoritmos baseados em DMM para cada valor de K.

Algoritmo	**Perplexidade para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
DMM	8.864	9.191	9.383	9.575	9.72	9.838	9.939	10.027	10.106	**10.176**
GPU-PDMM	8.746	9.085	9.287	9.435	9.554	9.653	9.741	9.819	9.887	**9.949**
NQTM	8.353	8.59	8.738	8.864	8.968	9.056	9.133	9.202	9.264	**9.319**
LapDMM	8.649	8.908	9.102	9.216	9.326	9.421	9.527	9.618	9.692	**9.791**

As perplexidades de todos os algoritmos aumentam consistentemente à medida que o número de tópicos cresce, como o DMM que varia de 8.864 com 10 tópicos até 10.176 para 100 tópicos. Este aumento na perplexidade é esperado, já que a incerteza na atribuição de palavras aos tópicos tende a aumentar com o maior número de tópicos.

Além disso, a diferença na perplexidade entre os algoritmos baseados em DMM tende a diminuir à medida que o número de tópicos aumenta. Isso pode ser devido à complexidade adicional na modelagem de mais tópicos, afetando a capacidade de todos os algoritmos de se diferenciarem significativamente.

A.2.2 Distância de tópicos

Os testes indicam que o LapDMM e NQTM apresentam consistentemente valores mais baixos de distância em comparação com os outros algoritmos, indicando maior presença de tópicos sobrepostos. A Figura 28 ilustra o desempenho de distância de tópicos de cada algoritmo.

A Tabela 11 mostra os valores de distância de tópicos para os algoritmos.

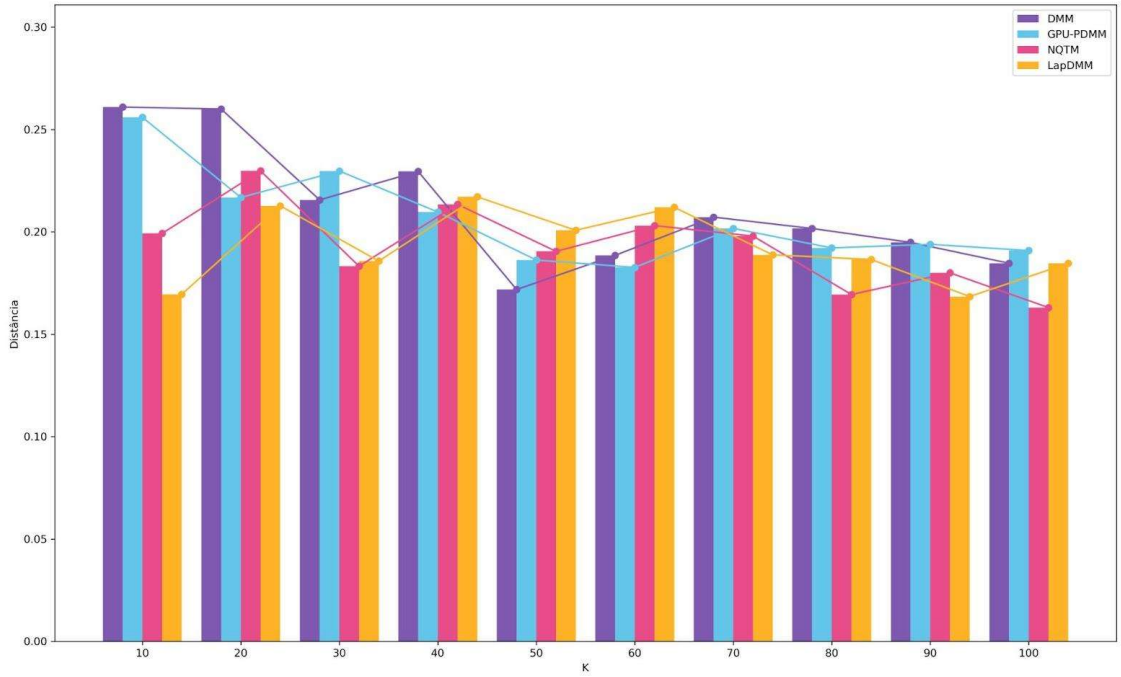
Tabela 11 – Distância de tópicos dos algoritmos baseados em DMM para cada valor de K.

Algoritmo	**Distância de tópicos para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
DMM	**0.261**	0.26	0.216	0.229	0.172	0.188	0.207	0.202	0.195	0.185
GPU-PDMM	**0.256**	0.217	0.23	0.21	0.186	0.183	0.202	0.192	0.194	0.191
NQTM	0.199	**0.23**	0.183	0.213	0.190	0.203	0.198	0.169	0.18	0.163
LapDMM	0.169	0.213	0.186	**0.217**	0.201	0.212	0.189	0.186	0.168	0.184

À medida que o número de tópicos aumenta, a distância de tópicos de todos os algoritmos tende a diminuir, indicando que a separação entre os tópicos se torna menos clara. Isso pode ser devido à crescente complexidade de diferenciar um número maior de tópicos nos documentos. Ademais, a variação na distância se torna menos pronunciada à medida que K aumenta.

Os algoritmos DMM e GPU-PDMM, em geral, apresentam valores de distância de tópicos maiores do que os algoritmos NQTM e LapDMM. Isso aponta que esses dois

Figura 28 – Distância de tópicos dos algoritmos baseados em DMM.



algoritmos podem ser mais adequados para cenários em que é desejável ter tópicos mais distintos e menos sobreposição entre os tópicos gerados. No entanto, também é importante considerar o contexto do problema ao escolher o algoritmo mais adequado, pois mais separação entre tópicos pode não ser necessariamente desejável em todas as situações.

A.2.3 Divergência-KL simétrica

Os algoritmos apresentam diferentes comportamentos em relação à divergência-KL simétrica e ao aumento do número de tópicos. O NQTM mostra ser mais eficiente em gerar tópicos distintos à medida que o número de tópicos aumenta, enquanto o LapDMM tende a gerar tópicos mais semelhantes em comparação com os outros algoritmos. A Figura 29 mostra os resultados de divergência de cada algoritmo.

A Tabela 12 mostra os valores da divergência-KL simétrica para os algoritmos.

Em geral, os valores de divergência para todos os algoritmos aumentam gradualmente à medida que o número de tópicos aumenta. Isso sugere que os modelos baseados em DMM conseguem gerar tópicos mais distintos à medida que K aumenta.

O GPU-PDMM apresenta valores próximos ao DMM, mas com uma tendência menos clara em relação ao aumento do número de tópicos. A divergência aumenta em alguns casos, mas não apresenta um padrão uniforme. Isso sugere que o GPU-PDMM pode gerar tópicos distintos, mas a relação entre o número de tópicos e a distinção dos tópicos gerados é menos previsível. O NQTM mostra-se mais eficiente em gerar

Figura 29 – Divergência-KL simétrica dos algoritmos baseados em DMM.

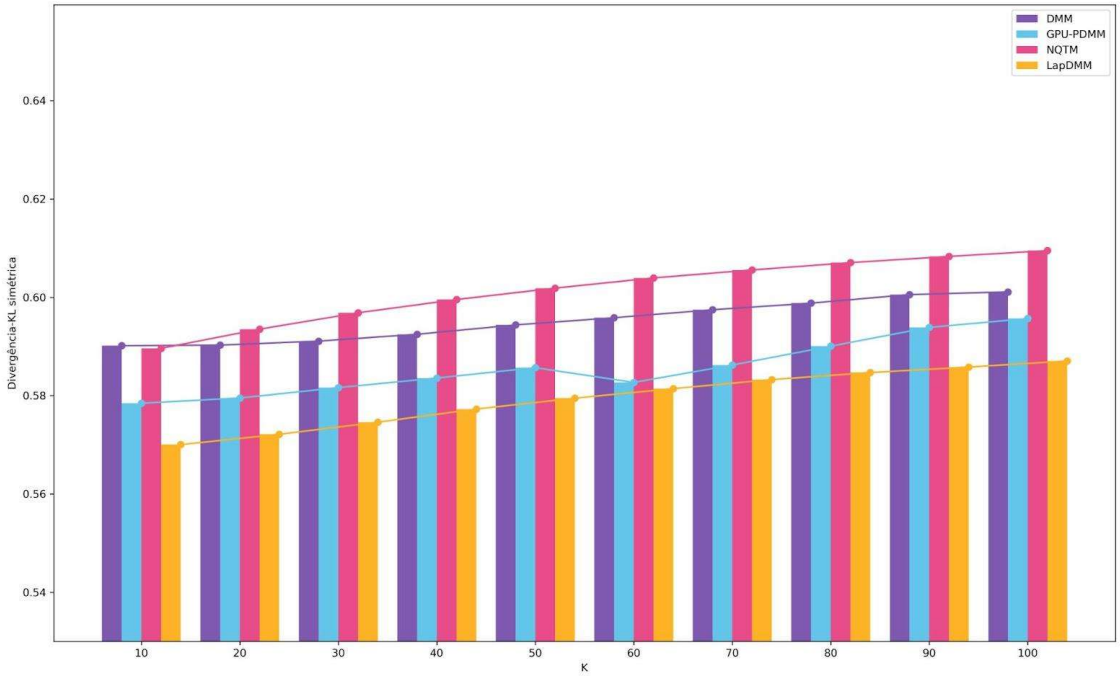


Tabela 12 – Divergência-KL simétrica dos algoritmos baseados em DMM para cada valor de K.

Algoritmo	**Divergência-KL simétrica para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
DMM	0.59	0.59	0.591	0.592	0.594	0.596	0.597	0.599	0.6	**0.601**
GPU-PDMM	0.578	0.579	0.582	0.584	0.586	0.583	0.586	0.59	0.594	**0.597**
NQTM	0.59	0.594	0.597	0.599	0.601	0.604	0.605	0.607	0.608	**0.609**
LapDMM	0.57	0.572	0.574	0.577	0.579	0.581	0.583	0.585	0.586	**0.587**

tópicos distintos com valores mais altos de K. Enquanto o LapDMM apresenta os valores mais baixos de divergência-KL simétrica entre os quatro algoritmos em todas as configurações de número de tópicos.

A.2.4 Coerência C_{UMASS}

Os resultados da análise apresentam diferentes níveis de coerência C_{UMASS}, com o DMM apresentando os valores mais altos, seguido pelo GPU-PDMM, LapDMM e, por último, o NQTM. A Figura 30 ilustra a coerência C_{UMASS} dos algoritmos baseados em DMM.

A Tabela 13 mostra os valores da coerência C_{UMASS} dos algoritmos.

O DMM apresenta os valores mais altos de coerência C_{UMASS} em todas as configurações de número de tópicos. Isso sugere que o DMM pode gerar tópicos mais coerentes, independentemente do número de tópicos. Em contrapartida, o NQTM

Figura 30 – Coerência C_{UMASS} dos algoritmos baseados em DMM.

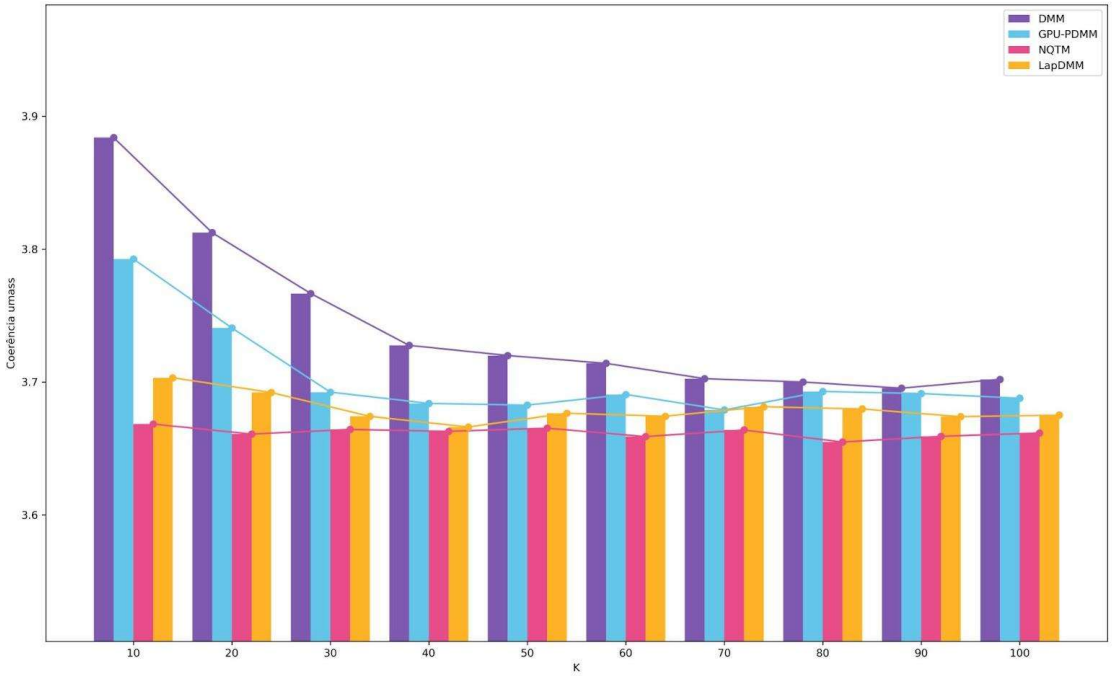


Tabela 13 – Coerência C_{UMASS} dos algoritmos baseados em DMM para cada valor de K.

Algoritmo	**Coerência C_{UMASS} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
DMM	**3.884**	3.812	3.767	3.728	3.72	3.714	3.702	3.7	3.695	3.702
GPU-PDMM	**3.793**	3.741	3.692	3.684	3.683	3.691	3.679	3.693	3.691	3.688
NQTM	**3.668**	3.661	3.664	3.663	3.665	3.659	3.664	3.655	3.659	3.662
LapDMM	**3.703**	3.692	3.674	3.666	3.676	3.674	3.681	3.68	3.674	3.675

tem os valores mais baixos entre os quatro algoritmos em todas as configurações de número de tópicos. Já o GPU-PDMM e LapDMM têm valores intermediários. Apesar disso, geralmente, as variações de coerência entre os algoritmos não são expressivas.

A.2.5 Coerência C_{W2V}

A avaliação indica que os algoritmos baseados em DMM podem gerar tópicos semanticamente coerentes. O LapDMM e o GPU-PDMM tendem a apresentar um desempenho ligeiramente superior em termos de coerência C_{W2V}, enquanto o DMM e o NQTM apresentam resultados ligeiramente inferiores. A Figura 31 ilustra a coerência C_{W2V} dos algoritmos baseados em DMM.

A Tabela 14 mostra os valores da coerência C_{W2V} dos algoritmos.

É possível observar, por meio dos resultados da coerência C_{W2V}, que os algoritmos baseados em DMM conseguem gerar tópicos semanticamente coerentes. Os

Figura 31 – Coerência C_{W2V} dos algoritmos baseados em DMM.

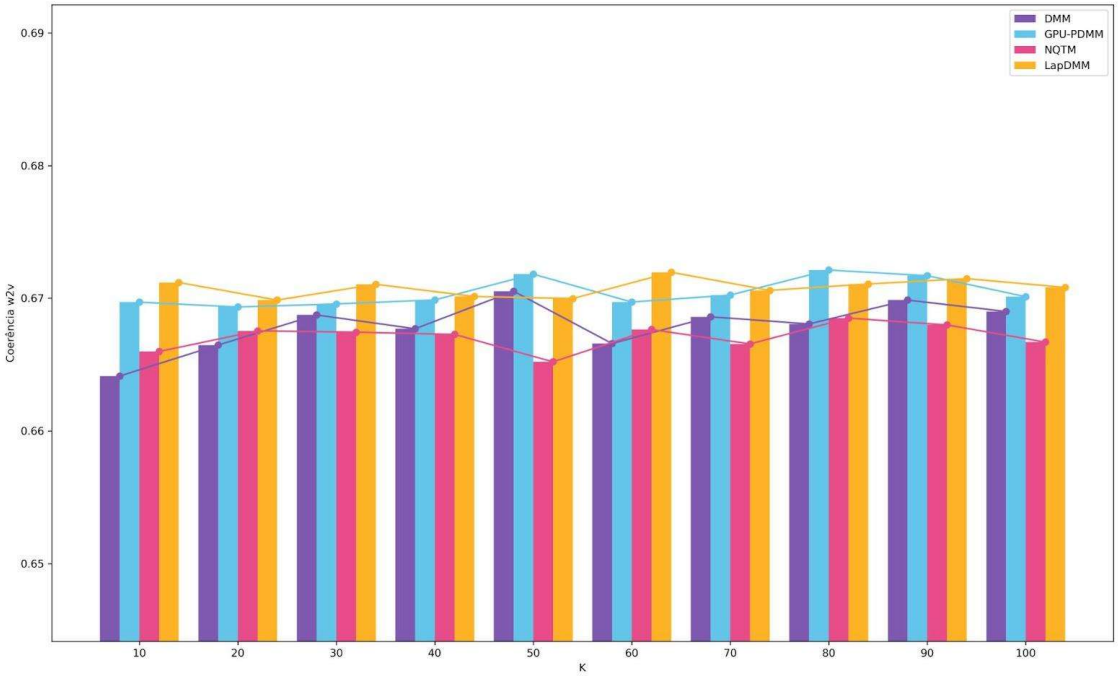


Tabela 14 – Coerência C_{W2V} dos algoritmos baseados em DMM para cada valor de K.

Algoritmo	**Coerência C_{W2V} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
DMM	0.664	0.666	0.669	0.668	**0.67**	0.666	0.669	0.668	**0.67**	0.669
GPU-PDMM	0.67	0.669	0.669	0.67	0.671	0.67	0.67	**0.672**	**0.672**	0.67
NQTM	0.666	**0.668**	0.667	0.667	0.665	**0.668**	0.667	**0.668**	**0.668**	0.667
LapDMM	0.671	0.67	0.671	0.67	0.67	**0.672**	0.67	0.671	0.671	0.671

resultados mostram que a variação entre os valores de C_{W2V} não é tão significativa, de forma que a coerência semântica dos tópicos é mantida conforme o número de tópicos aumenta. No entanto, é importante observar que, em alguns casos, o aumento de K resulta em uma ligeira melhora ou piora da coerência. Essa variação pode ser atribuída às características específicas dos dados e à interação entre o número de tópicos e a capacidade dos algoritmos em capturar a semântica subjacente.

A.3 BASEADOS EM AUTOAGREGAÇÃO (SATM, CRFTM, PTM, MIGA)

Os algoritmos baseados em autoagregação são uma classe de modelos de tópicos que buscam melhorar a qualidade e a interpretabilidade dos tópicos gerados, levando em utilizando do próprio conteúdo fornecido para a modelagem. Esses algoritmos usam mecanismos para autoagregar informações dos documentos fornecidos

para formar longos pseudo documentos, visando otimizar a modelagem de tópicos.

O SATM é um algoritmo de modelagem de tópicos que utiliza a ideia de autoagregação para identificar tópicos. Ele considera a relação entre palavras e a distribuição de tópicos nos documentos para criar grupos de palavras relacionadas.

O CRFTM é um modelo de tópicos que combina a modelagem de tópicos com CRF. O CRF é usado como um regularizador para abranger informações adicionais, como etiquetas de palavras ou relações entre palavras. Essa abordagem visa melhorar a interpretabilidade dos tópicos gerados ao levar em conta o contexto e as informações adicionais.

O PTM é um algoritmo de modelagem de tópicos que cria pseudo-documentos a partir de conjuntos de dados originais. Esses pseudo-documentos são então usados para treinar um modelo de tópicos. O PTM permite que os modelos de tópicos sejam treinados em dados que podem não ter uma estrutura clara de documento, como coleções de tuítes ou comentários. Essa abordagem ajuda a gerar tópicos melhores mesmo em conjuntos de dados com estruturas menos definidas.

O MIGA é um algoritmo de modelagem de tópicos que utiliza informações adicionais ou metainformações, como tags ou categorias, para orientar o processo de agregação. Essas informações são incorporadas no modelo para melhorar a coerência e a relevância dos tópicos gerados. Dessa forma, o grupo de algoritmos baseados em autoagregação foi analisado usando as 5 métricas de avaliação. A seguir, os resultados de cada análise são demonstrados e discorridos. Primeiramente será exibido um gráfico que ilustra o desempenho de cada algoritmo da categoria nos testes, e depois uma tabela com todos os resultados.

A.3.1 Perplexidade

A análise indica que o PTM e o CRFTM têm um desempenho melhor em termos de perplexidade, em comparação com o SATM e o MIGA. A Figura 32 ilustra o desempenho de perplexidade de cada algoritmo.

A Tabela 15 mostra em detalhes os valores da perplexidade de cada algoritmo. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

A perplexidade dos algoritmos aumenta à medida que o número de tópicos aumenta, o que é esperado, pois mais tópicos podem levar a uma maior complexidade no ajuste do modelo. No caso do SATM, o aumento da perplexidade é relativamente moderado, indicando que o modelo ainda pode lidar com mais tópicos eficientemente. A perplexidade do PTM apresenta valores menores do que os do SATM e CRFTM. Isso indica que a abordagem baseada em pseudo-documentos do PTM pode ser mais eficaz na criação de um modelo com boa generalização em comparação com as outras abordagens, especialmente para conjuntos de dados com estruturas menos definidas.

Figura 32 – Perplexidade dos algoritmos baseados em autoagregação.

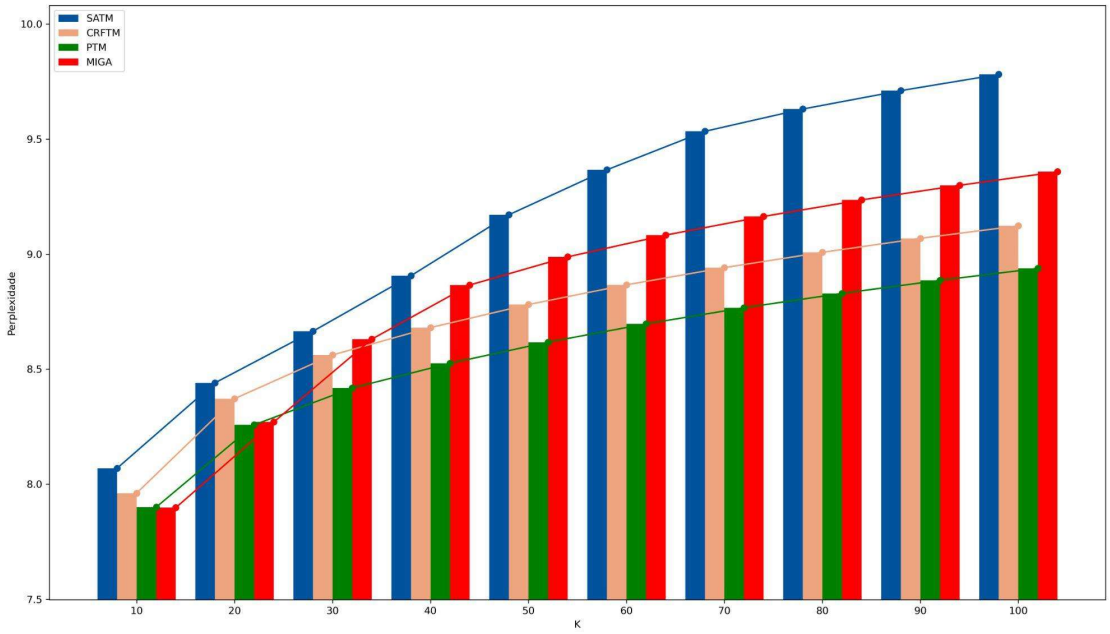


Tabela 15 – Perplexidade dos algoritmos baseados em autoagregação para cada valor de K.

Algoritmo	**Perplexidade para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
SATM	8.069	8.44	8.664	8.906	9.171	9.366	9.534	9.631	9.711	**9.781**
CRFTM	7.961	8.371	8.562	8.68	8.781	8.866	8.941	9.008	9.069	**9.123**
PTM	7.9	8.258	8.418	8.525	8.617	8.697	8.767	8.829	8.886	**8.938**
MIGA	7.897	8.27	8.63	8.865	8.988	9.083	9.164	9.235	9.299	**9.358**

A.3.2 Distância de tópicos

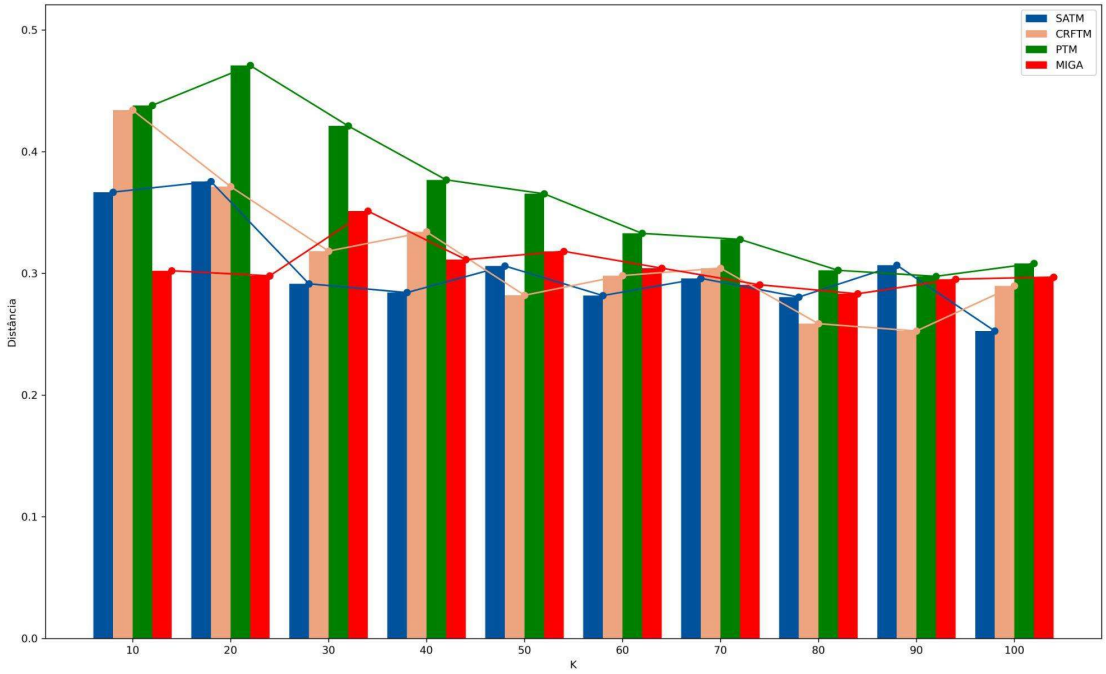
Os dados mostram que o PTM gera tópicos mais diversos, enquanto os outros algoritmos apresentam uma variação maior na distância de tópicos. A Figura 33 ilustra o desempenho de distância de tópicos de cada algoritmo.

A Tabela 16 mostra os valores de distância de tópicos para os algoritmos.

Tabela 16 – Distância de tópicos dos algoritmos baseados em autoagregação para cada valor de K.

Algoritmo	**Distância de tópicos para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
SATM	0.367	**0.375**	0.291	0.284	0.306	0.282	0.296	0.28	0.307	0.253
CRFTM	**0.434**	0.371	0.318	0.334	0.282	0.298	0.304	0.259	0.253	0.29
PTM	0.438	**0.471**	0.421	0.377	0.365	0.333	0.328	0.302	0.297	0.308
MIGA	0.302	0.298	**0.351**	0.311	0.318	0.304	0.291	0.283	0.295	0.297

Figura 33 – Distância de tópicos dos algoritmos baseados em autoagregação.



O CRFTM e SATM apresentam variações semelhantes na distância de tópicos, com valores tanto menores quanto maiores em diferentes valores de K. Isso aponta que os algoritmos são mais imprevisíveis, podendo gerar tópicos distintos em alguns casos, mas também podem gerar tópicos semelhantes em outros. O MIGA apresenta comportamento similar, porém com variação menor e mais estável. Quanto ao PTM, suas distâncias tendem a ser maiores em geral. Isso demonstra que o PTM pode capturar uma ampla gama de tópicos em diferentes níveis de granularidade, possivelmente devido à sua abordagem focada em pseudo-documentos.

A.3.3 Divergência-KL simétrica

A avaliação dos algoritmos mostra que, em geral, o CRFTM e o PTM tendem a gerar distribuições de tópicos mais semelhantes entre si, enquanto o SATM e o MIGA apresentam maior variação. Apesar disso, os valores de divergência de todos os algoritmos são muito similares com valores de K acima de 50. A Figura 34 ilustra o desempenho de divergência-KL simétrica de cada algoritmo.

A Tabela 17 mostra os valores da divergência-KL simétrica para os algoritmos.

A divergência do SATM é menos variável em todos os números de tópicos, com valores ligeiramente menores em comparação com os outros algoritmos. Já o CRFTM mostra uma divergência-KL simétrica ligeiramente mais alta em relação ao SATM, mas com uma tendência decrescente à medida que o número de tópicos aumenta. Além do mais, o MIGA varia em uma faixa semelhante à do CRFTM, com valores ligeiramente

Figura 34 – Divergência-KL simétrica dos algoritmos baseados em autoagregação.

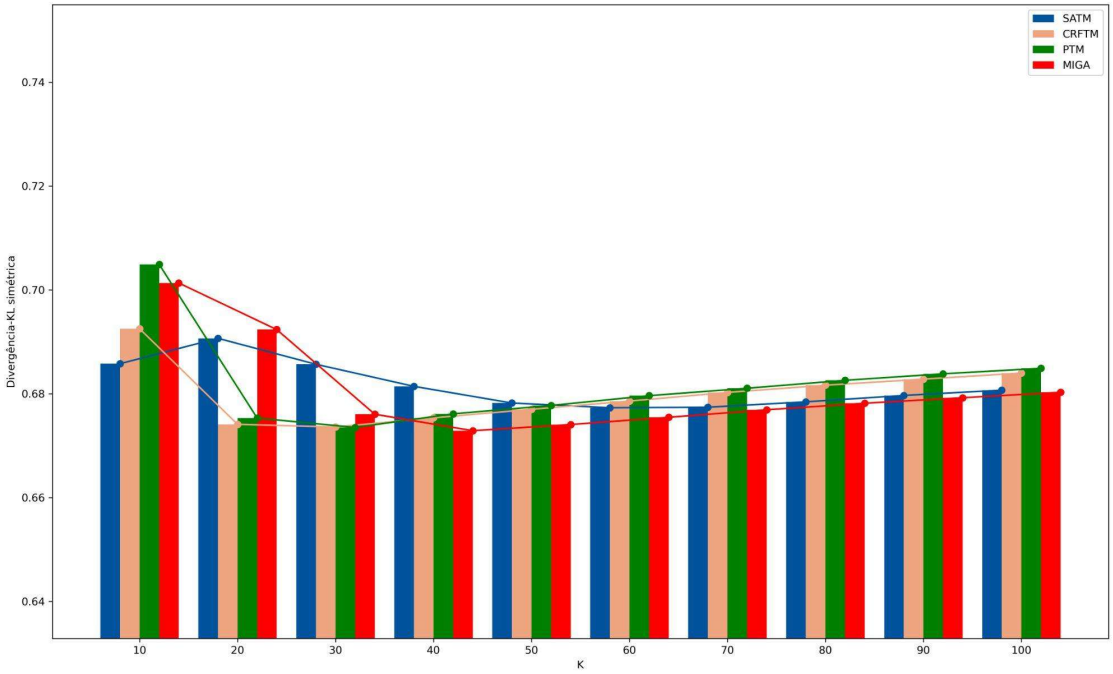


Tabela 17 – Divergência-KL simétrica dos algoritmos baseados em autoagregação para cada valor de K.

Algoritmo	**Divergência-KL simétrica para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
SATM	0.686	**0.691**	0.686	0.681	0.678	0.677	0.677	0.678	0.68	0.681
CRFTM	**0.692**	0.674	0.674	0.675	0.677	0.678	0.68	0.682	0.683	0.684
PTM	**0.705**	0.675	0.674	0.676	0.678	0.68	0.681	0.682	0.684	0.685
MIGA	**0.701**	0.692	0.676	0.673	0.674	0.675	0.677	0.678	0.679	0.68

mais altos em comparação com os outros algoritmos. O PTM começa com valores mais altos em comparação com os outros algoritmos, mas diminui à medida que K aumenta. Isso sugere que o PTM pode capturar uma gama mais ampla de tópicos.

A.3.4 Coerência C_{UMASS}

A análise indica que o SATM e o MIGA tendem a apresentar maior coerência nas palavras dos tópicos, enquanto o CRFTM e o PTM mostram valores mais baixos e estáveis de coerência. A Figura 35 ilustra a coerência C_{UMASS} dos algoritmos baseados em autoagregação.

A Tabela 18 mostra os valores da coerência C_{UMASS} para os algoritmos.

Os algoritmos apresentam um desempenho inicialmente alto, que diminui e se estabiliza à medida que o número de tópicos aumenta. O CRFTM apresenta valores

Figura 35 – Coerência C_{UMASS} dos algoritmos baseados em autoagregação.

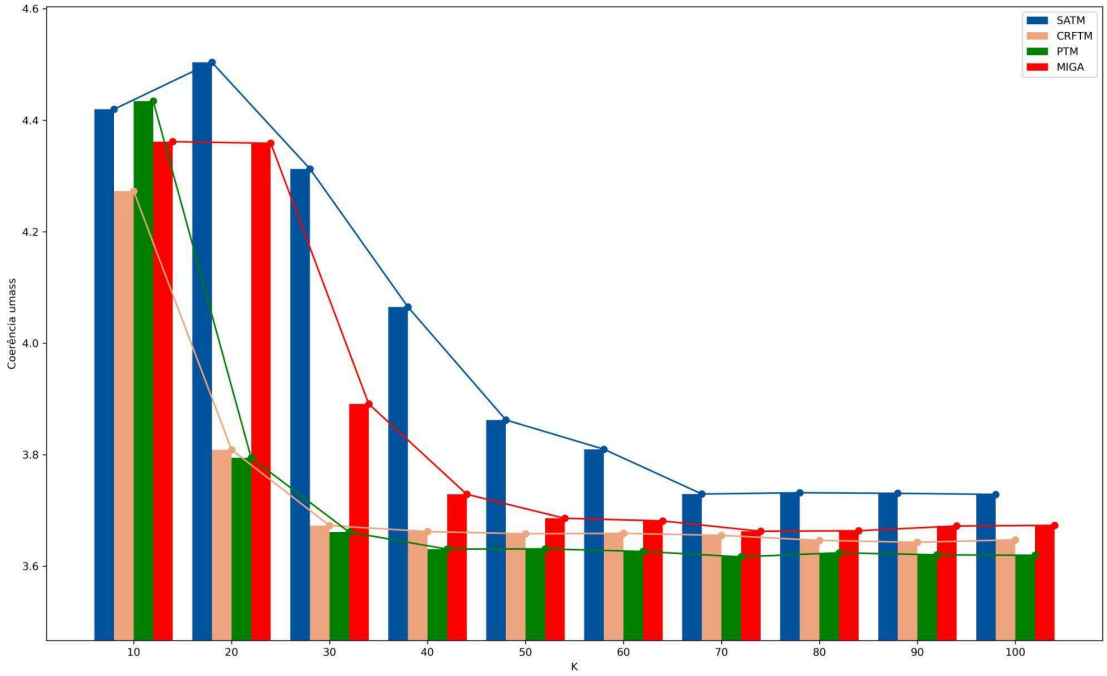


Tabela 18 – Coerência C_{UMASS} dos algoritmos baseados em autoagregação para cada valor de K.

Algoritmo	**Coerência C_{UMASS} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
SATM	4.42	**4.504**	4.313	4.065	3.862	3.81	3.729	3.732	3.731	3.729
CRFTM	**4.273**	3.809	3.673	3.662	3.658	3.659	3.655	3.646	3.643	3.647
PTM	**4.435**	3.794	3.661	3.631	3.631	3.626	3.617	3.624	3.62	3.619
MIGA	**4.362**	4.359	3.891	3.729	3.686	3.681	3.663	3.663	3.672	3.673

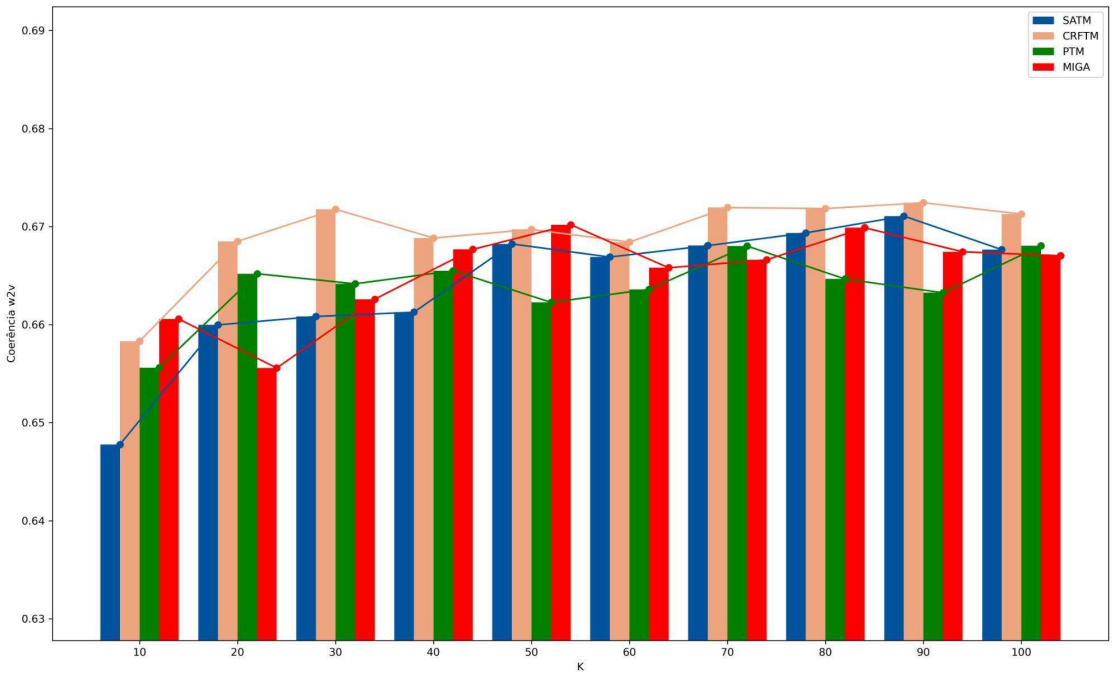
mais baixos em comparação com o SATM, e mostra pouca variação à medida que o número de tópicos aumenta. O PTM começa com uma coerência mais alta, e após o valor de K acima de 30, sua coerência se estabiliza no menor valor entre todos. Já o MIGA apresenta valores mais semelhantes ao SATM no início, e depois mostra uma tendência decrescente menos acentuada à medida que o número de tópicos aumenta.

A.3.5 Coerência C_{W2V}

De acordo com os dados, os algoritmos baseados em autoagregação apresentam desempenhos semelhantes em termos de coerência C_{W2V}, com valores que tendem a aumentar à medida que o número de tópicos cresce. CRFTM tem um desempenho levemente melhor que os outros, enquanto PTM tem um desempenho um pouco inferior. A Figura 36 ilustra a coerência C_{W2V} dos algoritmos baseados em

autoagregação.

Figura 36 – Coerência C_{W2V} dos algoritmos baseados em autoagregação.



A Tabela 19 mostra os valores da coerência C_{W2V} para os algoritmos.

Tabela 19 – Coerência C_{W2V} dos algoritmos baseados em autoagregação para cada valor de K.

Algoritmo	**Coerência C_{W2V} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
SATM	0.648	0.66	0.661	0.661	0.668	0.667	0.668	0.669	**0.671**	0.668
CRFTM	0.658	0.668	**0.672**	0.669	0.67	0.668	**0.672**	**0.672**	**0.672**	0.671
PTM	0.656	0.665	0.664	0.665	0.662	0.663	**0.668**	0.665	0.663	**0.668**
MIGA	0.661	0.655	0.663	0.668	**0.67**	0.666	0.667	0.67	0.667	0.667

O SATM apresenta um C_{W2V} relativamente estável, com um ligeiro aumento à medida que o número de tópicos aumenta. O CRFTM apresenta um desempenho semelhante ao SATM, com valores levemente mais altos e uma tendência de aumento à medida que K aumenta. Isso pode ser atribuído à inclusão de informações adicionais no modelo por meio de campos aleatórios condicionais, que ajudam a gerar tópicos mais coerentes. O PTM apresenta um desempenho ligeiramente inferior ao SATM e CRFTM, mas ainda mostra uma tendência de aumento na coerência à medida que o número de tópicos aumenta. Já o MIGA, apresenta um desempenho semelhante ao SATM, mas com valores de C_{W2V} ligeiramente mais baixos.

A.4 BASEADOS EM COOCORRÊNCIA GLOBAL (BTM, WNTM, MTM, NBTMWE)

Os algoritmos baseados em coocorrência global exploram as relações entre palavras que ocorrem juntas em um conjunto de documentos, ajudando a identificar tópicos subjacentes. Esses algoritmos consideram informações de coocorrência em todo o corpus, em vez de se concentrar apenas em informações locais, como a frequência das palavras em documentos individuais.

O BTM é um modelo de tópicos baseado em coocorrências globais que utiliza "biterms", pares de palavras coocorrentes em documentos, como unidades básicas de análise. Essa abordagem permite que o BTM capture relações de coocorrência em todo o corpus, melhorando a identificação de tópicos em comparação com métodos que se concentram apenas em informações locais.

O WNTM é um modelo que constrói uma rede de palavras para capturar as relações entre palavras coocorrentes. O WNTM identifica comunidades de palavras fortemente conectadas na rede e considera essas comunidades como tópicos. Essa abordagem aproveita a estrutura da rede para descobrir tópicos.

O MTM é um modelo de tópicos baseado em coocorrências globais que amplia a abordagem do BTM ao utilizar n-gramas, em vez de apenas *biterms*. Ao considerar sequências mais longas de palavras coocorrentes, o MTM pode capturar relações mais complexas entre palavras.

O NBTMWE é uma extensão do BTM que incorpora informações semânticas de *embeddings* de palavras, juntamente com *biterms*. Ao combinar a análise de coocorrência global com informações semânticas, o NBTMWE pode identificar tópicos considerando tanto a estrutura global do corpus quanto o significado das palavras.

Sendo assim, o grupo de algoritmos baseados em coocorrência global foi analisado usando as 5 métricas de avaliação. A seguir, os resultados de cada análise são demonstrados e discorridos. Novamente, será exibido um gráfico que ilustra o desempenho de cada algoritmo da categoria nos testes, e depois uma tabela com todos os resultados.

A.4.1 Perplexidade

A análise dos dados de perplexidade dos algoritmos baseados em coocorrência global revela que o MTM apresenta o melhor desempenho, seguido pelo NBTMWE, WNTM e, finalmente, BTM. A Figura 37 ilustra o desempenho de perplexidade de cada algoritmo.

A Tabela 20 mostra em detalhes os valores da perplexidade de cada algoritmo. Os maiores valores de cada algoritmo foram destacados com negrito, e os menores valores foram sublinhados.

Os algoritmos apresentam um aumento na perplexidade à medida que o número

Figura 37 – Perplexidade dos algoritmos baseados em coocorrência.

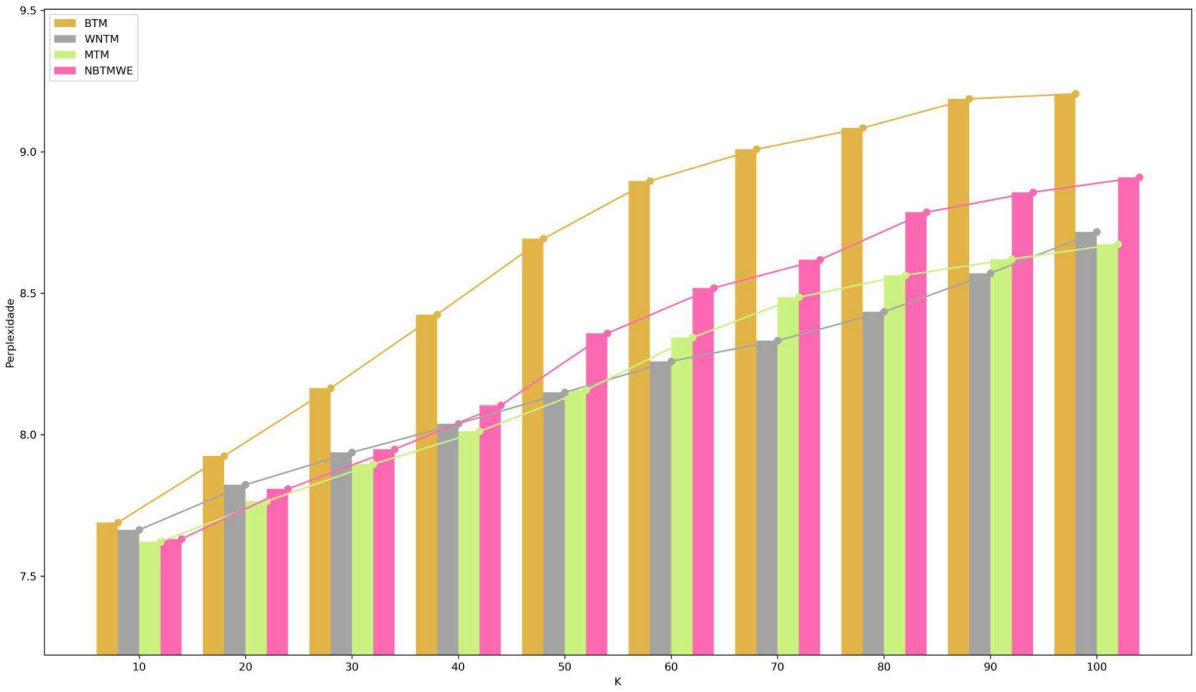


Tabela 20 – Perplexidade dos algoritmos baseados em coocorrência global para cada valor de K.

Algoritmo	**Perplexidade para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
BTM	7.689	7.925	8.164	8.425	8.693	8.897	9.009	9.084	9.187	**9.205**
WNTM	7.663	7.823	7.937	8.038	8.149	8.259	8.332	8.434	8.571	**8.716**
MTM	7.621	7.765	7.895	8.011	8.158	8.343	8.486	8.564	8.621	**8.674**
NBTMWE	7.631	7.808	7.948	8.104	8.358	8.519	8.619	8.786	8.857	**8.91**

de tópicos aumenta. O BTM apresenta um aumento mais acentuado, mostrando que os outros métodos são mais robustos para lidar com o aumento de K.

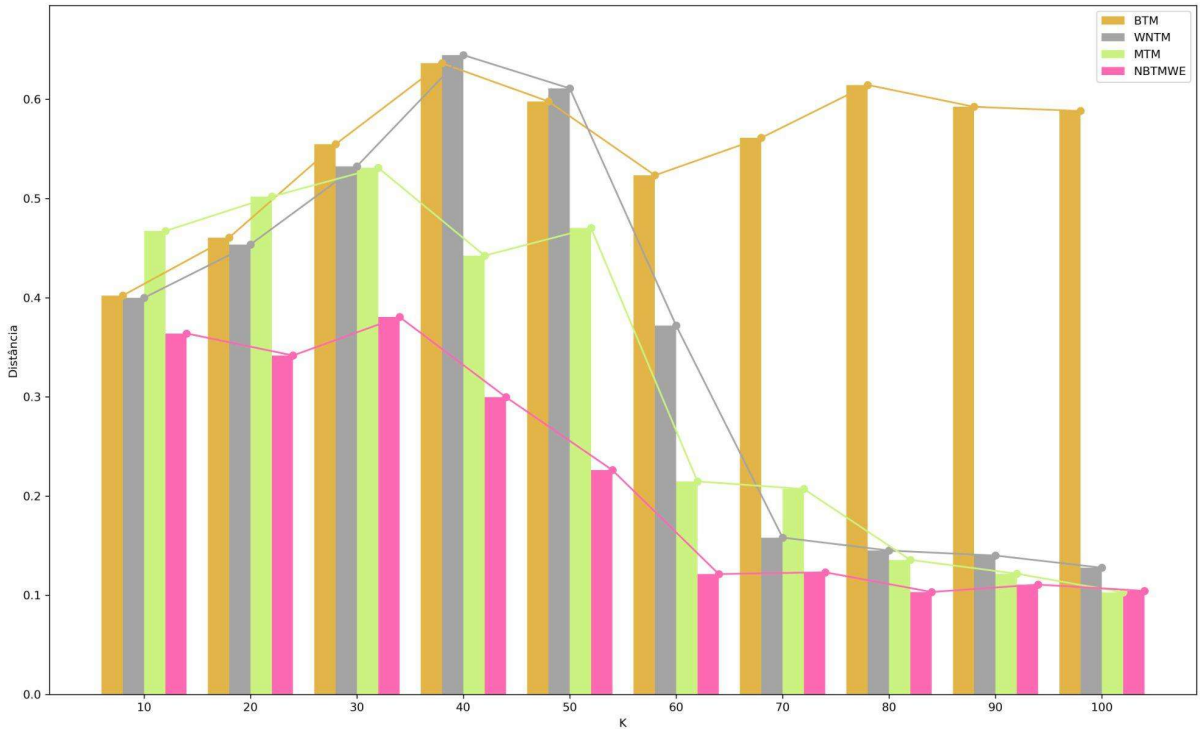
Pelos resultados, observa-se que não há um padrão claro de qual algoritmo possui a menor perplexidade em todas as configurações de número de tópicos, com MTM e WNTM alternando entre valores de perplexidade menores em várias configurações. Já o NBTMWE apresentam valores de perplexidade intermediários, especialmente quando o número de tópicos é maior.

A.4.2 Distância de tópicos

O BTM gera tópicos mais distintos consistentemente em diversos valores de K, enquanto os outros algoritmos mostram maior similaridade entre os tópicos à medida que o número de tópicos aumenta. A Figura 38 ilustra o desempenho de perplexidade

de cada algoritmo.

Figura 38 – Distância de tópicos dos algoritmos baseados em coocorrência.



A Tabela 21 mostra em detalhes os valores da distância de cada algoritmo.

Tabela 21 – Distância de tópicos dos algoritmos baseados em coocorrência global para cada valor de K.

Algoritmo	**Distância de tópicos para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
BTM	0.402	0.461	0.555	**0.636**	0.598	0.523	0.561	0.614	0.592	0.588
WNTM	0.4	0.454	0.532	**0.645**	0.611	0.372	0.158	0.145	0.140	0.128
MTM	0.467	0.502	**0.531**	0.442	0.47	0.214	0.207	0.136	0.122	0.103
NBTMWE	0.364	0.342	**0.381**	0.3	0.226	0.121	0.123	0.103	0.111	0.104

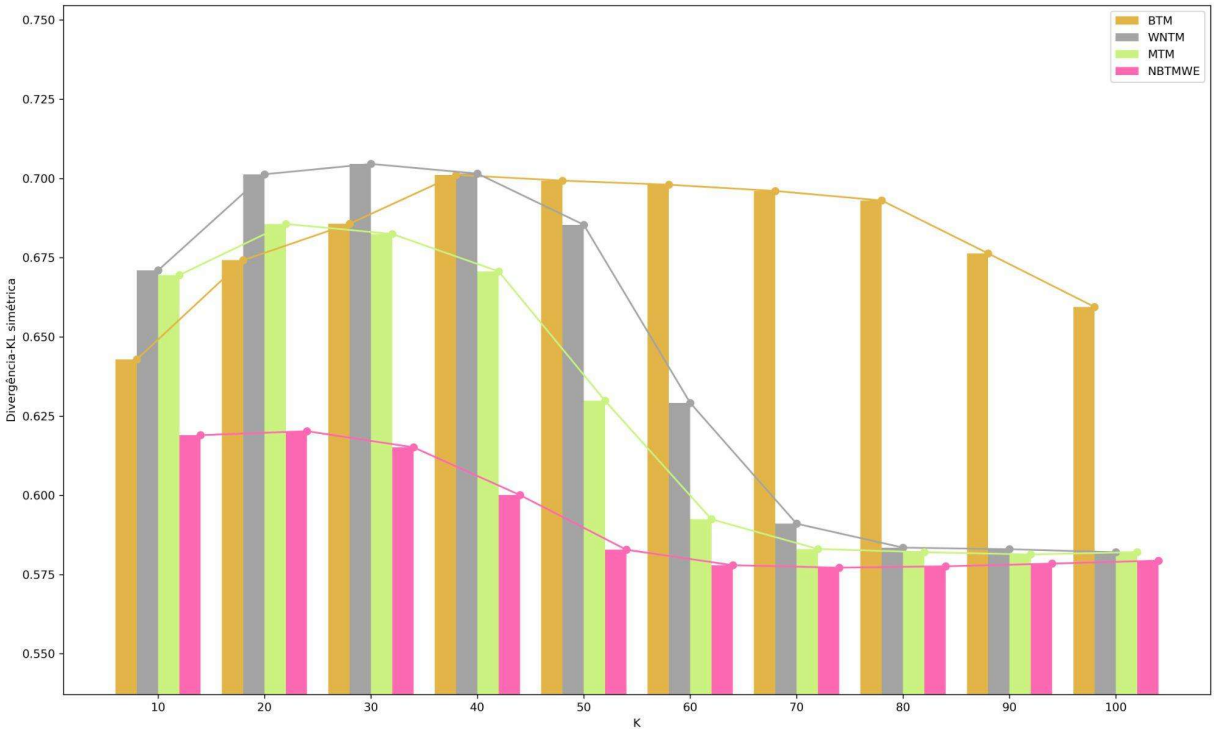
Os resultados mostram uma variação da distância de tópicos do BTM à medida que o número de tópicos aumenta. Não há uma tendência clara, mas a distância de tópicos permanece elevada. Já os outros algoritmos apresentam uma diminuição acentuada na distância de tópicos à medida que K aumenta.

Em valores de K mais baixos, o MTM apresenta os maiores resultados. No entanto, conforme se aumenta o número de tópicos, a distância diminui até chegar aos valores mais baixos entre todos os algoritmos. Similarmente, o NBTMWE apresenta atinge os menores valores já em 60 tópicos.

A.4.3 Divergência-KL simétrica

A análise dos dados mostra que o BTM e o MTM parecem gerar tópicos com distribuições mais distintas em um número menor de tópicos. Porém, juntamente do WNTM e o NBTMWE, o MTM apresentou maior similaridade entre as distribuições à medida que o número de tópicos aumentou. A Figura 39 mostra os resultados de divergência de cada algoritmo.

Figura 39 – Divergência-KL simétrica dos algoritmos baseados em coocorrência.



A Tabela 22 mostra em detalhes os valores da distância de cada algoritmo.

Tabela 22 – Divergência-KL simétrica dos algoritmos baseados em coocorrência global para cada valor de K.

Algoritmo	**Divergência-KL simétrica para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
BTM	0.643	0.674	0.686	**0.701**	0.699	0.698	0.696	0.693	0.676	0.659
WNTM	0.671	0.701	**0.704**	0.701	0.685	0.629	0.591	0.583	0.583	0.582
MTM	0.669	**0.686**	0.682	0.671	0.63	0.592	0.583	0.582	0.581	0.582
NBTMWE	0.619	**0.62**	0.615	0.6	0.583	0.578	0.577	0.578	0.578	0.579

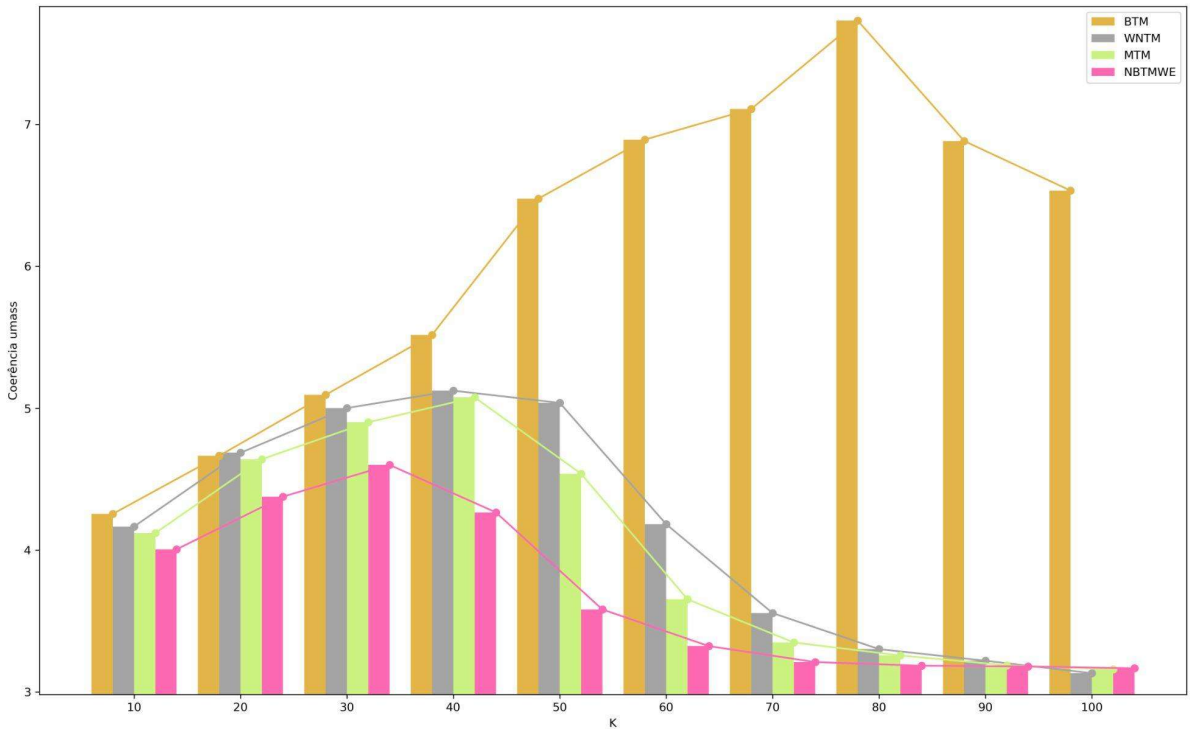
O BTM apresenta uma tendência geral de aumento na divergência-KL simétrica à medida que o número de tópicos aumenta, com uma ligeira queda nos últimos dois pontos (90 e 100 tópicos). Por sua vez, o WNTM mostra um padrão semelhante, com

um aumento inicial na divergência-KL simétrica à medida que o número de tópicos aumenta. Entretanto, o WNTM apresenta uma queda mais acentuada a partir de 50 tópicos. Já o MTM exibe um padrão semelhante ao WNTM, com um aumento inicial na divergência seguido por uma queda acentuada a partir de 40 tópicos. Por outro lado, o NBTMWE apresenta um padrão diferente dos outros modelos, com a divergência-KL simétrica diminuindo inicialmente e depois aumentando ligeiramente à medida que o número de tópicos aumenta.

A.4.4 Coerência C_{UMASS}

Os testes indicam que o BTM e o MTM geraram tópicos de maior qualidade em um número menor de K, enquanto o WNTM e o NBTMWE mostraram maior variação à medida que o número de tópicos aumentou. A Figura 40 ilustra o desempenho de coerência C_{UMASS} de cada algoritmo.

Figura 40 – Coerência C_{UMASS} dos algoritmos baseados em coocorrência.



A Tabela 23 mostra em detalhes os valores da coerência C_{UMASS} de cada algoritmo.

O BTM apresenta um aumento geral na coerência C_{UMASS} à medida que o número de tópicos aumenta, com um pico em 80 tópicos. No entanto, há uma diminuição nos dois últimos pontos (90 e 100 tópicos). Em contrapartida, os outros algoritmos exibem um padrão semelhante ao BTM, com um aumento inicial na coerência até cerca

Tabela 23 – Coerência C_{UMASS} dos algoritmos baseados em coocorrência global para cada valor de K.

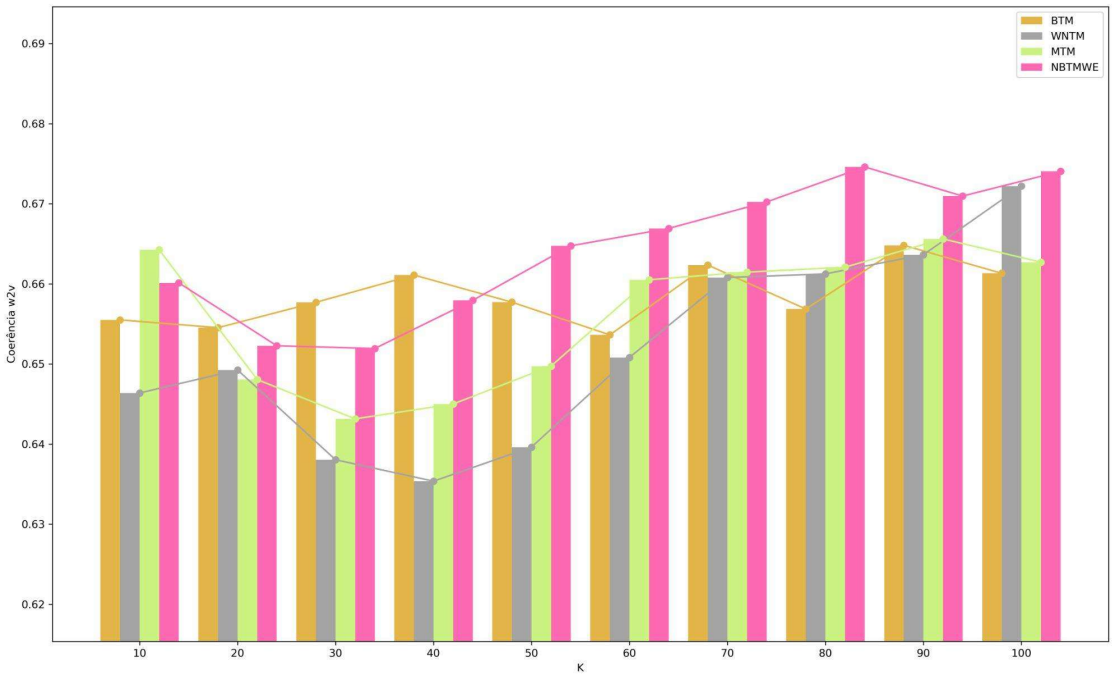
Algoritmo	**Coerência C_{UMASS} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
BTM	4.255	4.665	5.094	5.517	6.477	6.893	7.109	**7.731**	6.883	6.533
WNTM	4.166	4.686	5.001	**5.124**	5.038	4.183	3.557	3.303	3.221	3.133
MTM	4.121	4.641	4.901	**5.076**	4.539	3.654	3.349	3.258	3.188	3.16
NBTMWE	4.006	4.377	**4.601**	4.266	3.582	3.324	3.211	3.186	3.180	3.167

de 50 tópicos, seguido por uma queda acentuada. O NBTMWE apresenta um padrão semelhante, mas com a coerência diminuindo já em 40 tópicos e depois estabilizando à medida que o número de tópicos aumenta.

A.4.5 Coerência C_{W2V}

O BTM apresentou coerência C_{W2V} estável, com pouca variação em todos os números de tópicos. Por outro lado, o WNTM e o MTM mostraram uma queda inicial seguida de uma recuperação, se estabilizando nos valores maiores de K. Já o NBTMWE exibe uma tendência crescente na coerência. A Figura 41 ilustra o desempenho de coerência C_{W2V} de cada algoritmo.

Figura 41 – Coerência C_{W2V} dos algoritmos baseados em coocorrência.



A Tabela 24 mostra em detalhes os valores da coerência C_{W2V} de cada algoritmo.

Tabela 24 – Coerência C_{W2V} dos algoritmos baseados em coocorrência global para cada valor de K.

Algoritmo	**Coerência C_{W2V} para cada valor de K**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
BTM	0.655	0.654	0.658	0.661	0.658	0.654	0.662	0.657	**0.665**	0.661
WNTM	0.646	0.649	0.638	0.635	0.64	0.651	0.661	0.661	0.664	**0.672**
MTM	**0.664**	0.648	0.643	0.645	0.65	0.661	0.661	0.662	0.666	0.663
NBTMWE	0.66	0.652	0.652	0.658	0.665	0.667	0.67	**0.675**	0.671	0.674

Os dados mostram que a coerência C_{W2V} do BTM ficou relativamente estável em todos os números de tópicos. Há pequenas variações, mas nenhuma tendência clara. O WNTM e MTM se comportaram semelhantes entre si, com uma leve depressão inicial que logo se recupera e seguem estáveis a partir de 60 tópicos. Já o NBTMWE, em geral, apresentou um aumento na coerência à medida que o número de tópicos cresce.