UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Leandro da Silveira Dias

Mineração de padrões morfo-semânticos em textos literários com o BERT

Florianópolis
2022

Leandro da Silveira Dias

Mineração de padrões morfo-semânticos em textos literários com o BERT

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Sistemas de informação do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Sistemas de informação.
Orientador: Prof. Osmar de Oliveira Braz Junior, Me.
Coorientador: Prof. Renato Fileto, Dr.

Florianópolis
2022

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

Dias, Leandro da Silveirta
Mineração de padrões morfo-semânticos em textos literários com o BERT / Leandro da Silveirta Dias ; orientador, Osmar de Oliveira Braz Junior, coorientador, Renato Fileto, 2022.
56 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Sistema de Informação, Florianópolis, 2022.

Inclui referências.

1. Sistema de Informação. 2. Padrões Morfo-semânticos em Textos. 3. Embeddings. 4. Mineração de Textos. 5. Textos literários.. I. de Oliveira Braz Junior, Osmar. II. Fileto, Renato. III. Universidade Federal de Santa Catarina. Graduação em Sistema de Informação. IV. Título.

Leandro da Silveira Dias

Mineração de padrões morfo-semânticos em textos literários com o BERT

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Sistemas de informação e aprovado em sua forma final pelo curso de Graduação em Sistemas de informação.

Florianópolis, 2022.

Prof. Álvaro Junio Pereira Franco, Dr
Coordenador do Curso

Banca Examinadora:

Prof. Osmar de Oliveira Braz Junior, Me.
Orientador
Universidade Federal de Santa Catarina

Prof. Renato Fileto, Dr.
Coorientador
Universidade Federal de Santa Catarina

Prof. Ricardo Gaiotto de Moraes , Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Roberto Willrich , Dr.
Avaliador
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Aos meus orientadores. Ao Prof. Dr. Renato Fileto pela orientação, pelos conselhos e por ser um excelente mestre e modelo de didática e sabedoria. Ao Prof. Me. Osmar de Oliveira Braz Junior pela paciência, atenção e pelos muitos conhecimentos e experiencias compartilhados.

"Descobrir consiste em olhar para o que todo mundo está vendo e pensar uma coisa diferente” (Roger Von Oech)

RESUMO

Grande parte da informação atualmente disponível está na forma de textos, em documentos digitais como livros, artigos de jornais e revistas, páginas Web e textos em mídias sociais. O tratamento manual desses textos é frequentemente inviável, devido ao grande volume de dados, tornando-se necessário o desenvolvimento de soluções automatizadas para diversas tarefas de processamento de texto em linguagem natural. A análise semântica automatizada de discursos em torno de tópicos de interesse em documentos textuais é um problema ainda em aberto, com diversas aplicações práticas, incluindo detecção de certos tipos de discurso (e.g. discurso de ódio) e classificação não supervisionada de textos com base em similaridades e padrões semânticos dos discursos neles encontrados. Este trabalho se propõe a desenvolver novas técnicas e algoritmos para minerar padrões morfo-semânticos de discursos centrados em tópicos do interesse de especialistas de domínio. Tais tópicos podem ser mencionadas nos textos literalmente (através das palavras que os definem) ou via léxicos que tenham semântica equivalente ou muito próxima a tais tópicos. A implementação do protótipo utiliza *embeddings* do BERTimbau, uma versão do modelo contextualizado de linguagem BERT pré-treinada para o português brasileiro atual, como meio de determinar similaridades semânticas entre palavras, que podem indicar tópicos de interesse ou expressar a semântica dos discursos em torno de tais tópicos. Ferramentas de Processamento de Linguagem Natural (PLN) existentes também são utilizadas para realizar tarefas como segmentação de sentenças, normalização de texto (e.g., tokenização, *stemming*) e classificação morfossintática de palavras (*Part-Of-Speech - POS-tagging*). Os algoritmos sendo desenvolvidos para minerar padrões morfo-semânticos em textos se apoiam nas funcionalidades dessas ferramentas, principalmente similaridade semântica entre *embeddings* contextualizados de palavras e compatibilidade de PoS-tags. A proposta está sendo desenvolvida e avaliada em um estudo de caso na área de literatura brasileira, em que especialistas de domínio fornecem os textos a analisar, indicam os tópicos de interesse e auxiliam na aferição dos resultados. Os resultados serão avaliados quantitativamente, em termos da distribuição das instâncias dos padrões minerados nas coleções de documentos e, na medida das possibilidades, comparação com o desempenho humano na identificação dos padrões e classificação dos textos.

Palavras-alvo: Padrões Morfo-semânticos em Textos. Processamento de Linguagem Natural (PLN). Mineração de Textos. Embeddings. Textos literários.

ABSTRACT

Much of the information currently available is in the form of texts, in digital documents such as books, newspaper and magazine articles, web pages and texts in social media. The manual treatment of these texts is often not feasible, due to the large volume of data, making it necessary to develop automated solutions for various text processing tasks in natural language. Automated semantic analysis of discourses around topics of interest in textual documents is an open problem, with several practical applications, including detection of certain types of speech (eg hate speech) and unsupervised classification of texts based on similarities and semantic patterns of the discourses found in them. This work proposes to develop new techniques and algorithms to mine morphosemantic patterns of discourses centered on topics of interest to domain experts. Such topics can be mentioned in texts literally (through the words that define them) or via lexicons that have equivalent semantics or very close to such topics. The prototype implementation uses BERTimbau embeddings, a version of the pre-trained BERT language contextualized model for current Brazilian Portuguese, as a means of determining semantic similarities between words, which may indicate topics of interest or express the semantics of the discourses around of such topics. Existing Natural Language Processing (NLP) tools are also used to perform tasks such as sentence segmentation, text normalization (eg, tokenization, stemming) and morphosyntactic word classification (Part-Of-Speech - POS-tagging). Algorithms being developed to mine morpho-semantic patterns in texts are based on the functionalities of these tools, mainly semantic similarity between contextualized embeddings of words and compatibility of PoS-tags. The proposal is being developed and evaluated in a case study in the area of Brazilian literature, in which domain specialists provide the texts to be analyzed, indicate the topics of interest and help in the assessment of the results. The results will be quantitatively evaluated, in terms of the distribution of instances of mined patterns in document collections and, as far as possible, comparison with human performance in identifying patterns and classifying texts.

Keywords: Morpho-semantic Patterns in Texts. Natural Language Processing (NLP). Text Mining. Embeddings. Literary texts.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BERT	Bidirectional Encoder Representationsfrom Transformers
BLPL	Biblioteca Digital de Literatura de Países Lusófonos
DS	*Data Scientist*
EL	*Entity Linking*
GloVe	*Global Vectors for Word Representation*
GPU	*Graphics Processing Unit*
GSDMM	*Gibbs Sampling for the Dirichlet Multinomial Mixture*
MCL	Modelo Contextualizado de Linguagem
MSC	*Componentes Morfo-semânticos*
NED	*Named Entities Disambiguation*
NER	*Named Entities Recognition*
NPMI	*Normalized Pointwise Mutual Information*
PCA	*Principal Component Analysis*
PLN	Processamento de Linguagem Natural
POS	*Part-Of-Speech*
SNE	*t-Distributed Stochastic Neighbor Embedding*
SSP	*Short Semantic Pattern*
TM	*Text Mining*
TPU	*Tensor Processing Unit*
UMAP	*Uniform Manifold Approximation and Projection for Dimension Reduction*
WSD	*Word Sense Disambiguation*
WSI	*Word Sense Induction*

SUMÁRIO

1 INTRODUÇÃO

O consumo de informação na forma de texto em meio digital vem aumentando rapidamente. Segundo o jornal Folha de São Paulo Porto (2020), houve um aumento de 115% na venda de *E-Books* entre os anos de 2016 a 2019. Mais recentemente, a pandemia do COVID-19 potencializou o consumo desses textos digitais. O grande volume de textos torna difícil a análise manual dos seus conteúdos, o que tem levado ao desenvolvimento de soluções utilizando ferramentas para Processamento de Linguagem Natural (PLN), mineração de textos (do inglês, *Text Mining* (TM)) e ciência de dados (do inglês, *Data Scientist* (DS)).

Técnicas e ferramentas de PLN têm sido utilizadas com sucesso em diversas tarefas, incluindo reconhecimento de entidades nomeadas (do inglês, *Named Entities Recognition* - NER) para encontrar menções a entidades relacionadas à COVID-19 e auxiliar na resposta à pandemia (SHORTEN; KHOSHGOFTAAR; FURHT, 2021), ligação de entidades (do ingles, *Entity Linking*) para vincular entidades na área de biomedicina (CHEN; VAROQUAUX; SUCHANEK, 2021) e classificação morfossintática de palavras (do inglês, *Part-Of-Speech* - POS-*Tagging*) que, juntamente com embeddings de palavras e aprendizado de máquina, permite detectar e analisar discursos de ódio em textos de mídias sociais (SORATO; GOULARTE; FILETO, 2020; LEITE et al., 2020). Atualmente, *embeddings* contextualizados têm possibilitado ganhos consideráveis de desempenho em tarefas como categorizar textos de portais de noticias (SHENG; YUAN, 2021), analisar coerência em textos (BRAZ-JUNIOR; FILETO, 2021) e sumarizar textos longos (SHENG; YUAN, 2021).

1.1 DESCRIÇÃO DO PROBLEMA

A Biblioteca Digital de Literatura de Países Lusófonos (BLPL)[1], além de oferecer acesso a metadados e ao próprio texto de um grande número de obras literárias em língua portuguesa, é um exemplo de coleção de textos digitais sobre a qual são desenvolvidas, exercitadas e avaliadas estratégias para ensino de literatura, análise literária e leitura distante. Seu acoplamento ao Moodle e à ferramenta de anotação semântica DLNotes (WILLRICH et al., 2020), entre outras, permite realizar, por exemplo, tarefas de anotação semântica de textos. Este ambiente constitui um excelente campo experimental de novas tecnologias para mineração e análise de textos, incluindo análise semântica de discursos. O Núcleo de Pesquisa em Informática, Literatura e Linguística (NUPILL-UFSC)[2], que desenvolveu e mantém a BLPL e o DLNotes, entre outros artefatos, inclui especialistas em literatura, linguagem humana e computação, que unem esforços para inovar na confluência dessas áreas, além de colaboradores diversos para auxiliar na definição de soluções e avaliação do seu desempenho. Ao serem apresentados à possibilidades de efetuar análise semântica automatizada de discursos mediante mineração de padrões em torno de tópicos fornecidos, os especialistas em literatura e linguística do NUPILL sugeriram

[1] https://www.literaturabrasileira.ufsc.br/public/index.php
[2] https://nupill.ufsc.br/

como primeira tarefa para estudo de caso em literatura minerar padrões semânticos de discursos em torno do tópico ***escravidão*** (que pode ser expresso por palavras como *escravidão, escravo, negro, mulato* e *senzala*) em obras selecionadas de alguns autores.

O problema tratado neste trabalho, em colaboração com pesquisadores do NUPILL, é minerar padrões semânticos de discurso em torno de tópicos fornecidos. Um padrão semântico refere-se a um conjunto de instâncias de componentes textuais (e.g. sentenças) com palavras de sentido similar em torno de palavras que definam o tópico focado. A título de exemplo, as seguintes sentenças, extraídas da obra *As Vítimas-Algozes*, de Joaquim Manuel de Macedo, publicada em 1869, constituem instâncias do mesmo padrão semântico em torno do tópico ***escravidão***, o qual também pode é denotado pela palavra ***escravo*** nessas sentenças:

1. *"Em falta de pundonor*[1] *e de vergonha*[2]*, que a* ***escravidão*** *não comporta, o* ***escravo*** *tem o rancor*[3] *e o desejo da vingança*[4]*."*

2. *"Nas pontas do açoite*[1] *está o emblema do rancor*[3] *do* ***escravo****: às vezes há nas pontas do açoite*[2] *marcas de sangue*[4]*."*

O padrão semântico de discurso é determinado pela compatibilidade semântica entre palavras usadas em torno do tópico ***escravidão*** nas duas sentenças acima (instâncias do padrão). A palavra *pundonor*[1] da primeira sentença pode ser considerada semanticamente relacionada à *açoite*[1] na segunda sentença, assim como *vergonha*[2] a *açoite*[2] e *vingança*[4] a *sangue*[4]. Além disso, a palavra *rancor*[3] está em ambas as sentenças. Note que o discurso em torno do tópico pode incluir um número distinto de palavras (com significado relevante) em cada sentença, cada uma dessas palavras pode ser compatível com várias palavras da outra sentença e não precisa haver correspondência biunívoca entre palavras das duas sentenças para haver o padrão. Este padrão é definido pelo alinhamento semântico dos discursos das duas sentenças, que pode ser medido como similaridade entre palavras, possivelmente mas não necessariamente considerando também a compatibilidade de suas classes morfo-sintáticas e/ou morfologia, pois também podem influenciar no sentido das palavras, assim como o contexto onde elas são usadas. Assim, podemos denominar tais padrões como morfo-semânticos.

1.2 OBJETIVOS

O objetivo geral deste trabalho é minerar padrões morfo-semânticos em textos literários, visando suportar classificação não supervisionada e análise semântica de discursos em torno de tópicos fornecidos, em um estudo de caso na área de literatura. Para alcançá-lo, é necessário atingir os objetivos específicos abaixo relacionados.

1. Estudar, selecionar e dominar técnicas e ferramentas do estado da arte Processamento de Linguagem Natural (PLN) necessárias para minerar padrões morfo-semânticos em textos.

2. Desenvolver e avaliar novos algoritmos eficientes e efetivos para minerar padrões morfo-semânticos em torno de tópicos fornecidas por usuários especialistas de domínio usando técnicas e ferramentas de PLN estudadas e selecionadas.
3. Analisar a distribuição das instâncias de padrões, classes morfossintáticas e sentidos das palavras envolvidas em textos literários, visando classificação e análise semântica de discursos de acordo com as ocorrências de tais padrões.

1.3 METODOLOGIA

Inicialmente, serão realizados estudos sobre o estado da arte em tarefas e ferramentas de PLN que podem ser usadas na preparação dos dados textuais a serem minerados, nos próprios algoritmos de mineração de padrões morfo-semânticos e na avaliação dos seus resultados. Tais tarefas incluem normalização de texto (tokenização, *stemming*, etc.), classificação morfossintática (*PoS-Tagging*), reconhecimento de entidades nomeadas, ligação de entidades e cálculo de similaridade entre *embeddings* de palavras. Posteriormente, serão realizadas análises qualitativas e quantitativas das distribuições de palavras presentes nos textos a serem minerados, suas classes morfossintáticas, distâncias entre seus respectivos *embeddings* e outras medidas. Tais análises visam preparação adequada dos dados e definição de parâmetros para mineração de padrões morfo-sintáticos, incluindo, entre outros, seleção de tópicos e palavras que os representem, além de funções de similaridade ou distância semântica e limiares a serem considerados na determinação de palavras compatíveis.

Para resolver o problema de minerar automaticamente padrões morfo-semânticos, podem ser utilizadas diversos alternativas de medidas de similaridade e critérios adicionais para determinar compatibilidade de palavras. Neste trabalho exploramos *embeddings* contextualizados de palavras para determinar similaridade (e portanto compatibilidade) semântica de palavras, além de seus *POS-Tags* (e.g., substantivo, verbo), nos algoritmos que estamos desenvolvendo, adaptando e avaliando para minerar automaticamente os padrões morfo-semânticos em torno de tópicos fornecidos por usuários especialistas do domínio de literatura. Descrições desses artefatos são apresentadas em detalhes nos capítulos 2, 3 e 4.

Finalmente, será realizada a avaliação dos custos computacionais e da qualidade dos resultados obtidos com os algoritmos desenvolvidos. Será preparado um artigo com descrições dos algoritmos desenvolvidos com links para código-fonte no GitHub e demonstrações de seu funcionamento em notebooks do Google Colab, além das tarefas efetuadas nos experimentos, dados utilizados, configurações de parâmetros e análise dos resultados obtidos. A distribuição das instâncias dos padrões minerados nas coleções de documentos será avaliada quantitativamente, com o objetivo de caracterizar textos de diferentes autores, épocas e movimentos. Os custos computacionais serão avaliados em termos de tempo de processamento e uso de memória. Além disso, na medida das possibilidades, os resultados dos algoritmos serão confrontados com a apreciação humana na identificação dos padrões, classificação dos textos e análise semântica dos discursos.

1.4 ESTRUTURA DO TRABALHO

O restante deste trabalho está estruturado como se segue. O Capítulo 2 descreve os fundamentos utilizados no trabalho e necessários ao seu entendimento. O Capítulo 3 discute os trabalhos relacionados e compara as características do trabalho aqui proposto com o estado-da-arte. O Capítulo 4 descreve a proposta para minerar padrões morfo-semânticos em textos literários, cujo objetivo é propiciar meios para analisar e classificar tais de acordo com a semântica dos discursos expressa nesses padrões. O Capítulo 5 delineia o plano de experimentos para avaliar a proposta, reporta experimentos iniciais e discute seus resultados. Finalmente, o Capítulo 6 apresenta as conclusões parciais, o cronograma de atividades para finalizar o trabalho e enumera temas para trabalhos futuros.

2 FUNDAMENTOS

Este capítulo primeiro fornece uma visão geral de Processamento de Linguagem Natural (PLN), incluindo conceitos fundamentais, aplicações e ferramentas. Em seguida, descreve as tarefas de PLN utilizadas no desenvolvimento da solução proposta neste trabalho. Posteriormente, discute *embeddings* de palavras, os principais modelos disponíveis para gerá-los, suas propriedades, técnicas de *pooling* para compor embeddings de componentes textuais maiores como sentenças, e cálculo de funções de distância e similaridade usando *embeddings*. O foco é no modelo contextualizado de linguagem BERT e a sua variação *BERTimbau* pré-treinada para o português brasileiro atual. Finalmente, discute a principais funcionalidades da ferramenta *Embedding projector*, utilizada para visualizar *embeddings* manipulados neste trabalho.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

Processamento de Linguagem Natural (PLN), segundo Jurafsky e Martin (2009), é um ramo da inteligência artificial que visa dar ao computador a capacidade de processar linguagem humana. Esse campo de estudo é interdisciplinar e por esse motivo ele pode ser denominado por outros nomes, tais como: processamento de fala e linguagem, tecnologia da linguagem, linguística computacional e reconhecimento e síntese de fala. O objetivo é fazer com que os computadores possam executar tarefas envolvendo linguagem humana, possibilitando a comunicação humano-máquina, melhorando a comunicação entre humanos ou simplesmente fazendo processamento útil de texto e/ou fala. Chowdhary (2020) define PLN como um conjunto de métodos computacionais para realizar análise automática e representação das linguagens humanas. Algumas das aplicações mais comuns de PLN são: classificação de texto, tradução automática, extração de informações, aquisição de conhecimento, sumarização automática de textos e resposder perguntas estipuladas em linguagem natural (em inglês *Question Answering - QA*).

Atualmente há diversos kits de ferramentas para PLN, tais como: *spaCy*[1], *Stanza*[2] *FreeLing*[3] e *LX-Center*[4]. Tarefas de PLN suportadas por tais kits incluem tokenização, *Part-Of-Speech - POS-tagging*), *stemming* ou lemmatização, segmentação de sentenças (em inglês *Sentence Boundary Detection* - SBD), reconhecimento de entidades nomeadas (em inglês *Named Entity Recognition - Named Entities Recognition (NER)), ligação de entidades (em Entity Linking - Entity Linking (EL) ou Named Entities Disambiguation (NED) (Disambiguation)) e determinação de similaridades.* Este trabalho usa o *spaCy*, uma biblioteca de código aberto e gratuita com métodos e modelos pré-treinados em língua portuguesa para realizar tarefas como as citadas anteriormente. As subseções a seguir descrevem as de tarefas PLN usadas no desenvolvimento deste trabalho.

[1] https://spacy.io/
[2] https://stanfordnlp.github.io/stanza/
[3] https://nlp.lsi.upc.edu/freeling/
[4] http://lxcenter.di.fc.ul.pt/

Normalização de texto

A normalização de texto é uma fase de pre-processamento de texto que engloba a realização de várias tarefas do PLN visando padronizar palavras do *corpus* a ser analisado. Essa padronização pode incluir tarefas como converter o texto para letra maiúscula ou minúscula e remoção de acentos, caracteres especiais e repetições. A tarefa de normalização de texto visa tornar a sua análise mais precisa, por remover possíveis ruídos no momento da análise estatística ou do modelo utilizado. Como exemplo, considere o texto a seguir extraído do livro *As Vítimas-Algozes*, de Joaquim Manuel de Macedo, seguido de sua normalização.

Trecho original: *"Nas pontas do açoite está o emblema do rancor do escravo: às vezes há nas pontas do açoite marcas de sangue"*

Trecho normalizado: *"nas pontas do acoite esta o emblema do rancor do escravo: as vezes ha nas pontas do acoite marcas de sangue"*

Note que no trecho normalizado, ocorreu a conversão de todas as letras maiúsculas para minuscula (e.g "Nas"para "nas"), bem como a remoção dos acentos (e.g "está"para "esta") e caracteres especiais (e.g "açoite"para "acoite"). Repetições (e.g "...."para ".") também são removidas quando ocorrem no texto original.

Tokenização

A tokenização separa os *tokens* (palavras e sinais representativos) do texto. Métodos de tokenização aplicam regras ou modelos que variam com o idioma. O exemplo a seguir mostra o texto do exemplo anterior com cada token sublinhado sublinhados.

Trecho tokenizado " nas pontas do acoite esta o emblema do rancor do escravo : as vezes ha nas pontas do acoite marcas de sangue "

Stemming ou lematização

As tarefas de *Stemming* e lematização servem para reduzir cada palavra à sua forma raiz, reduzindo o ruído devido a flexos (gênero, número, conjugação verbal, etc.). A lematização retorna o lema da palavra, (e.g. o lema de "pontas"é "ponta"), enquanto a tarefa de *Stemming* retorna o seu radical chamado em inglês *stem* (e.g. o radical de "pontas"é "pont"). A Tabela 1 lista lemas e radicais dos tokens do exemplo anterior.

Detecção de Limites de Sentenças

A tarefa de Detecção de Limites de Sentenças (SBD), ou simplemente segmentação de sentenças, realiza o processo que quebra o texto em sequências de sentenças conforme a sinais

Tabela 1 – Comparação entre *Lemmatization* e *Stemming*

Token	**Lema**	**Radical (*Stem*)**
Nas	em o	na
pontas	ponta	pont
do	de o	do
açoite	açoite	acoit
está	estar	est
o	o	o
emblema	emblema	emblem
do	de o	do
rancor	rancor	ranc
do	de o	do
escravo	escravo	escrav
:	:	:
às	a o	as
vezes	vez	vez
há	haver	ha
nas	em o	na
pontas	ponta	pont
do	de o	do
açoite	açoite	acoit
marcas	marca	marc
de	de	de
sangue	sangue	sang

Fonte: o autor.

de pontuação, como o ponto final. A título de exemplo, consideremos outro fragmento de texto do livro *As Vítimas-Algozes*, de Joaquim Manuel de Macedo a seguir:

Trecho original *"Excelente crioulo! Como ama a seu senhor! Há poucos assim. As aparências dissimulavam os sentimentos do escravo."*

A tarefa de segmentação de sentenças divide este texto texto em 4 sentenças:

1. - Excelente crioulo!"
2. "Como ama a seu senhor!"
3. "Há poucos assim."
4. "As aparências dissimulavam os sentimentos do escravo."

PoS-Tagging

A anotação morfossintática das palavras, usualmente denotada pelo termo em inglês *Part-Of-Speech Tagging* ou abreviadamente *POS-Tagging*, categoriza cada token de um corpus de acordo com sua classe morfossintática (verbo, adjetivo, etc.). A Tabela 2 apresenta as classes

Tabela 2 – Tabela de rótulos PoS-*Tagging*

Rótulo	**Classe Gramatical**
X	Outro
VERB	Verbo
SYM	Símbolo
CONJ	Conjunção
SCONJ	Conjunção subordinativa
PUNCT	Pontuação
PROPN	Nome próprio
PRON	Pronome substantivo
PART	Partícula, morfemas livres
NUM	Numeral
NOUN	Substantivo
INTJ	Interjeição
DET	Determinante (artigo e pronomes adjetivos)
CCONJ	Conjunção coordenativa
AUX	Verbo auxiliar
ADV	Advérbio
ADP	Preposição
ADJ	Adjetivo

Fonte: o autor.

Tabela 3 – Tarefa de *POS-Tagging* em sentença

Token	***POS-Tag***	**Token**	***POS-Tag***
Nas	ADP	:	ADP
pontas	NOUN	às	NOUN
do	ADP	vezes	VERB
açoite	NOUN	há	ADP
está	AUX	nas	NOUN
o	DET	pontas	ADP
emblema	NOUN	do	NOUN
do	ADP	açoite	PROPN
rancor	NOUN	marcas	ADP
do	ADP	de	NOUN
escravo	NOUN	sangue	

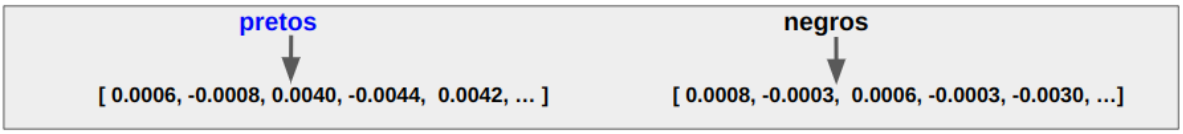
Fonte: o autor.

morfossintáticas utilizadas pela ferramenta spaCy. Diferentes ferramentas podem utilizar diferentes conjuntos de classes morfossintáticas, os quais podem ser mais ou menos detalhados. Métodos e modelos para realizar *POS-Tagging* usualmente levam em consideração o contexto onde os tokens ocorrem no texto ou fala para tentar atribuir corretamente uma etiqueta/rótulo de classe morfossintática a cada token. Segundo Sorato et al. (2016), é de grande importância que seja precisa a classificação dos elementos morfossintáticos de uma sentença. *POS-Tags* errôneos podem ocasionar erros de processamento subsequentes, porque diversas outras tarefas de PLN e mineração de texto dependem de *POS-Tagging* correto. A Tabela 3 apresenta o resultado da aplicação do método de *POS-Tagging* do SpaCy ao texto do exemplo anterior.

2.2 EMBEDDINGS

Um *embedding* de palavra pode ser definido a grosso modo como uma representação de uma palavra por meio de um vetor usualmente com centenas de dimensões. Cada dimensão de um tal vetor contém um número real, geralmente, entre -1 e 1 (CORDEIRO, 2019). A representação vetorial é criada de tal modo que palavras com sentido similares ficam próximas umas das outras no espaço vetorial multidimensional, entre outras propriedades que podem ser capturadas. A Figura 1 ilustra *embeddings* de duas palavras distintas mas cuja semântica pode ser considerada similar, dependendo do contexto em que são usadas: “pretos"e “negros".

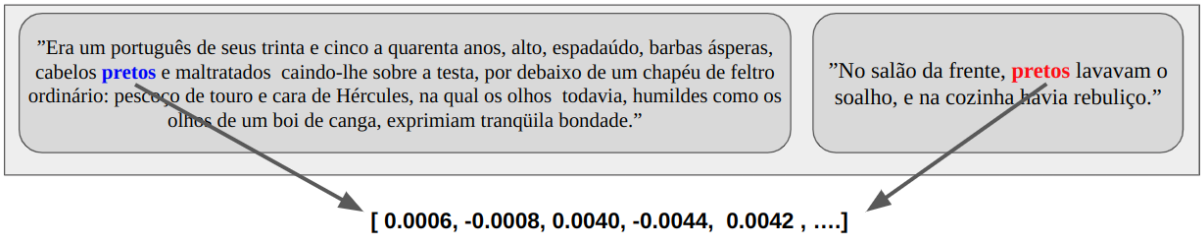
Figura 1 – Palavras e seus *embeddings*



Fonte: o autor.

Os modelos de *embedding* mais tradicionais são *Word2Vec* (MIKOLOV et al., 2013) e *Global Vectors for Word Representation* (GloVe) (ZHANG; ZHAO; WANG, 2020). Muitas tarefas de PLN têm se beneficiado do seu uso. Porém, esses modelos tradicionais têm uma única representação vetorial para cada palavra, mesmo que seus significado ou mesmo classe morfossintática mude em diferentes contextos onde é usada, como ilustrado na Figura 2.

Figura 2 – *Embeddings* de uma mesma palavra em frases diferentes



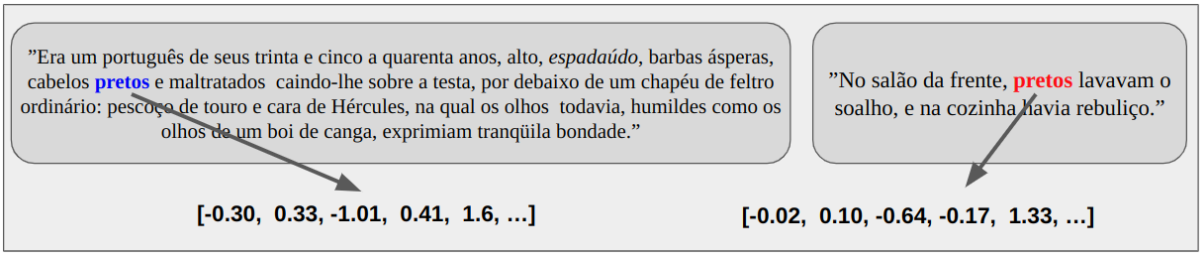
Fonte: o autor.

Na Figura 2 é possível ver que a palavra **"pretos"** tem sentidos distintos nas duas sentenças. Na sentença da esquerda, tal palavra faz o papel de adjetivo e tem o sentido de cor, enquanto na sentença da direta o sentido faz papel de substantivo com o sentido de indivíduo humano. Assim, as nuances de significado não são capturadas na representação vetorial. Modelos contextualizados de linguagem atuais permitem contornar este problema gerando diferentes *embeddings* para um mesmo token, pois consideram o contexto onde as palavras estão inseridas.

2.2.1 Modelos Contextualizados de Linguagem

Estudos recentes procuram meios de realizar a compreensão da comunicação em linguagem natural de forma automática através da aplicação de Modelo Contextualizado de Linguagem (MCL) (do inglês, *Contextualized Language Model*) (ZHANG; ZHAO; WANG, 2020). Os atuais MCLs costumam se baseados em rede neural bidirecional que possibilita capturar o contexto em que uma palavra ocorre em um texto, e assim o seu significado específico naquele contexto. MCLs permitem ajustar *embeddings* de um mesmo léxico de acordo com o contexto, gerando representações diferentes para uma mesma palavra de modo a capturar nuances de significado em diferentes contextos de uso, como ilustrado pela Figura 3.

Figura 3 – Palavras e seus *embeddings* contextualizados



Fonte: o autor.

BERT: Bidirectional Encoder Representations from Transformers

O BERT (acrônimo do inglês, *Bidirectional Encoder Representations from Transformers*) (DEVLIN et al., 2019) é um MCL constituído de uma rede neural profunda com processamento bidirecional. Ele é pré-treinado em um corpus de texto que ultrapassa 33 milhões de documentos textos não rotulados de cunho geral. Porém, permite ajustes finos com a adição de apenas uma camada de saída, visando otimizar o seu desempenho em tarefas de classificação de texto, reconhecimento de entidades nomeadas, em certos tipos de corpus como textos em blogs, diálogos entre tutores e estudantes (DEVLIN et al., 2019). O BERT apresenta ótimos resultados na descoberta de estrutura sintática e semântica dentro de uma sentença (TENNEY; DAS; PAVLICK, 2019) em língua inglesa. Embeddings contextualizados como os gerados pelo BERT são úteis na solução de diversas tarefas de PLN, incluindo sumarização, classificação de textos e sistemas de diálogo. O BERT também tem propiciado ganhos consideráveis de desempenho quando ajusta pra efetuar tarefas de PLN, tais como segmentação (MULLER; BRAUD; MOREY, 2019) e comparação semântica de sentenças (REIMERS; GUREVYCH, 2019) e classificação de documentos (OSTENDORFF et al., 2020).

Trabalhos anteriores do nosso grupo de pesquisa usaram embeddings não contextualizados para a mineração de padrões morfo-semânticos em textos curtos, tais como postagens em mídias sociais (SORATO et al., 2016; SORATO; GOULARTE; FILETO, 2020), visando

apoiar a análise de discursos. Textos longos, tais como os de literatura, têm mais informação de contexto. Assim, *embeddings* contextualizados têm potencial de contribuir para a qualidade da mineração de padrões morfo-semânticos nesses textos. Desta forma, este trabalho investiga o uso do BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020), uma versão do BERT pré-treinada para língua portuguesa na mineração padrões morfossintáticos nas obras literárias. Este trabalho gera *embeddings* contextualizados usando o *BERTimbau* e os utiliza para calcular distâncias ou similaridades em algoritmos de mineração de padrões morfo-semânticos em textos. O BERTimbau foi escolhido devido à disponibilidade gratuita de seus modelos pré-treinados, inclusive em língua portuguesa, e por conveniência do seu uso no *Google Colaboratory*, que disponibiliza acesso direto aos modelos através de bibliotecas específicas. O BERTimbau, assim como o BERT original, está disponível em dois tamanhos: $BERTimbau_{base}$ com 12 níveis e 110 milhões de parâmetros e $BERTimbau_{large}$ com 24 níveis e 335 milhões de parâmetros.

2.2.2 Distâncias e similaridade entre *embeddings*

A mineração de padrões morfo-semânticos que podem estar presentes em textos literários requer determinar elementos (palavras, trechos curtos em torno de palavras) semanticamente próximos uns com os outros. As métricas (VALLERIAN, 2021) de distância e similaridade utilizadas neste trabalho são apresentadas a seguir.

Distância Euclidiana

A Equação 2.1 determina a distância Euclidiana (também conhecida com L2) entre dois vetores. Esta métrica indica o quão próximos dois pontos (x,y) estão em um plano, tomando o comprimento do caminho mais curto entre eles. A Figura 4 ilustra a distância euclideana entre dois no plano [5].

$$dist_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

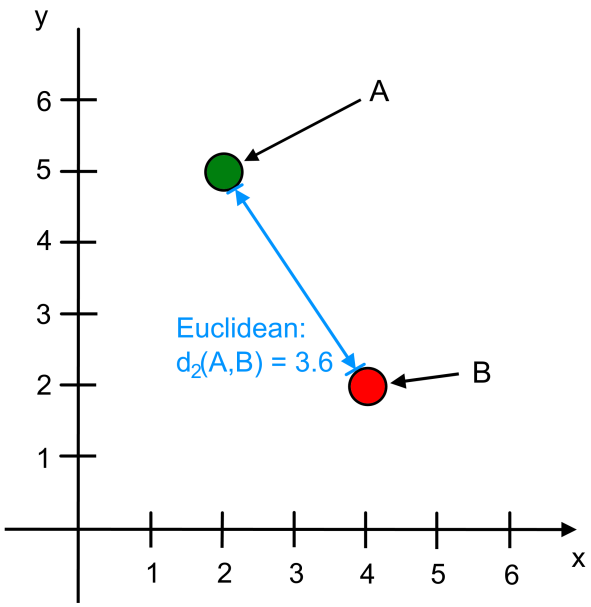
Distância de Manhattan

A distância de Manhattan conforme Equação 2.2 (também conhecida como L1) é a distância entre dois pontos (x,y) medidos ao longo de eixos em ângulos retos. Devido aos ângulos retos também é chamada distância *City-Block*. A Figura 5 ilustra a L2 entre dois dois pontos de um plano.

$$dist_{man}(x,y) = \sum_{i=1}^{n}|(x_i - y_i)| \tag{2.2}$$

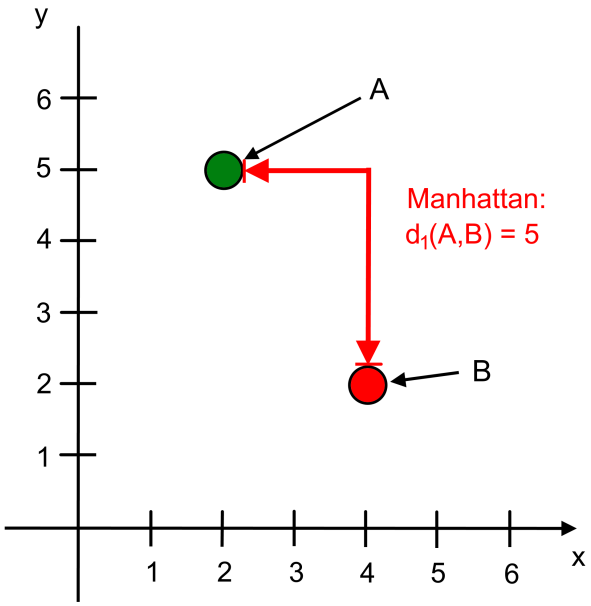
[5] Esta e as duas figuras seguintes estão disponíveis em https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa

Figura 4 – Distância Euclidiana



Fonte: Adaptada de What... (2020).

Figura 5 – Distância Manhattan



Fonte: baseado em What... (2020)

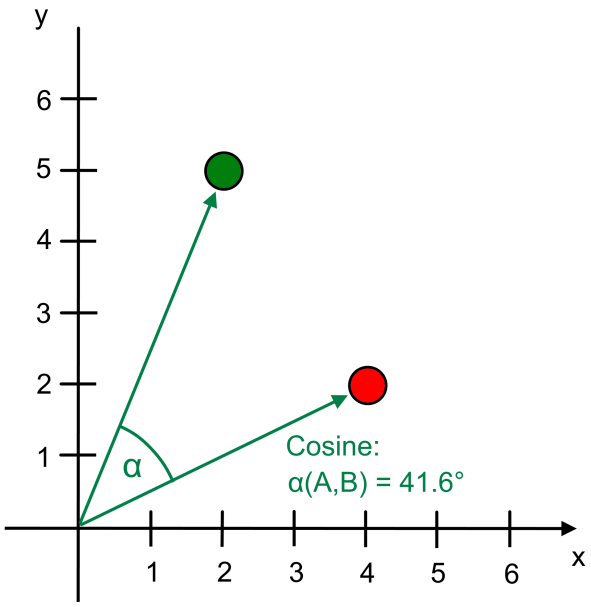
Similaridade de cosseno

A Equação 2.3 determina a similaridade cosseno, que difere das anteriores por variar no intervalo $[0, 1]$ sendo 0 correspondente a distância infinita e 1 correpondente a distância 0. Esta medida de similaridade avalia o valor do cosseno do ângulo compreendido entre dois

vetores (A,B) num espaço vetorial. A Figura 6 ilustra a similaridade cosseno entre dois no plano.

$$sim_{cos}(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.3}$$

Figura 6 – Similaridade de cosseno



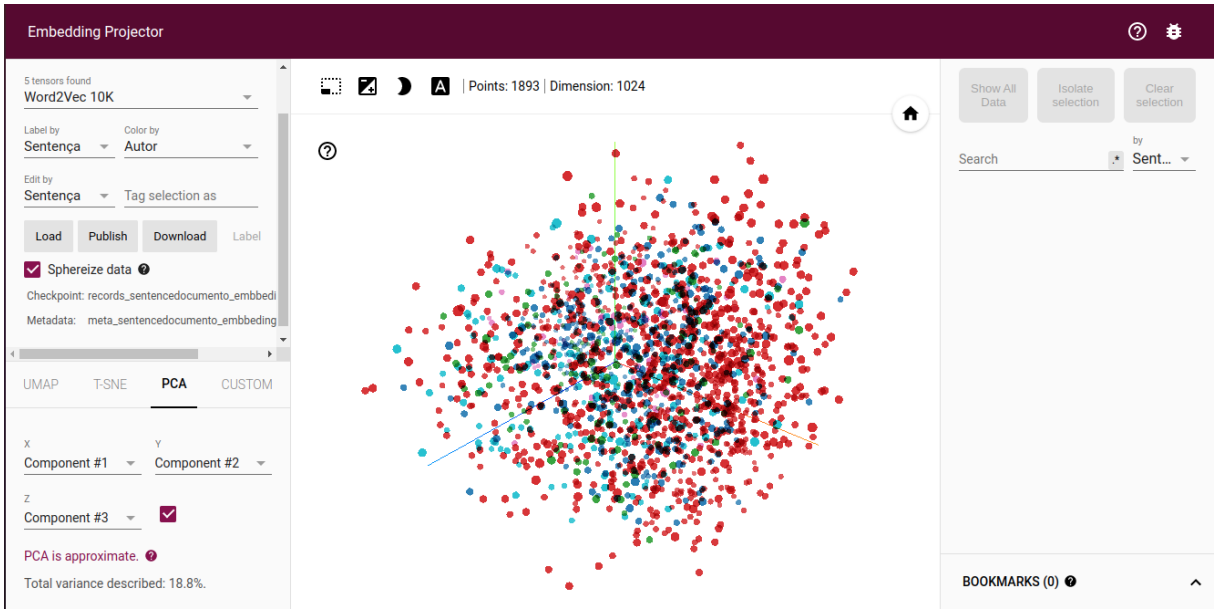
Fonte: baseado em What... (2020)

Além de calcular a distância e a similaridade entre os palavras é possível fazê-lo também com trechos curtos de textos. Para tal, pode-se concatenar os *embeddings* das palavras contidas no trecho que se queira comparar para obter um *embeddings* do trecho inteiro.

2.3 EMBEDDING PROJECTOR

O *Embedding Projector* (SMILKOV et al., 2016) é uma ferramenta de visualização de *embeddings* em duas ou três dimensões que ajuda a interpretar modelos de aprendizado de máquina que dependem de *embeddings*. A ferramenta permite explorar a vizinhança de pontos representando *embeddings* individuais, analisar a distribuição global dos pontos e investigar vetores semanticamente significativos no espaço. Possibilita realizar análises visuais das distribuições dos *embeddings* e buscas em formato de texto para testar hipóteses. O *Embedding Projector* é implementado como uma aplicação Web sobre a plataforma do *TensorFlow* para visualizar qualquer conjunto de *embeddings*, de qualquer dimensionalidade, fornecido através da plataforma ou em formato texto. A Figura 7 ilustra a sua interface gráfica.

Figura 7 – *Embedding Projector*



Fonte: o autor.

2.3.1 Métodos para reduzir a dimensionalidade de um conjunto de dados

O *Embedding Projector* oferece quatro métodos para reduzir a dimensionalidade de um conjunto de dados: dois lineares e dois não linear. Cada um desses métodos, descritos a seguir, pode ser usado para criar uma visão bi ou tridimensional.

Análise do componentes principais (PCA)

Análise do componentes principais (do inglês, *Principal Component Analysis* - PCA) (JOLLIFFE, 1986) é uma técnica multivariada que pode ser usada para reduzir a dimensionalidade de dados. PCA utiliza uma transformação ortogonal para converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas chamadas de componentes principais. o PCA pode ser usado para fornecer uma visualização em dimensões mais baixas dos mesmos dados. Isto é feito usando-se apenas os primeiros componentes principais. O *Embedding Projector* calcula os 10 principais componentes principais e permite ao usuário escolher dentre esses componentes em qualquer combinação de dois ou três dimensões para efetuar a projeção dos dados. O PCA é frequentemente eficaz para visualizar a geometria global e encontrar *clusters*.

Incorporação de vizinhos estocásticos distribuídos t (T-SNE)

Incorporação de vizinhos estocásticos distribuídos t (do englês, t-Distributed Stochastic Neighbor Embedding - *t-SNE*) (MAATEN; HINTON, 2008) é uma técnica de redução de

dimensionalidade não linear. O projetor oferece a Incorporação da visualizações *t-SNE* bidimensionais e tridimensionais. A geração da visualização é executado do em tempo real utilizando os recursos disponível no computador do usuário. Como o *t-SNE* geralmente preserva alguma estrutura local, na prática ele suporta tanto de na tarefa de identificar os vizinhos mais próximo dos pontos quanto a visualização da geometria global e encontrar *clusters*.

Aproximação e Projeção de Manifold Uniforme para Redução de Dimensões (UMAP)

O t-SNE é uma excelente técnica para visualizar conjuntos de dados de alta dimensão. Porém, apresenta algumas desvantagens, tais como o tempo de computação elevado e perda de informações em grande escala. Aproximação e Projeção de Manifold Uniforme para Redução de Dimensões (do inglês *Uniform Manifold Approximation and Projection for Dimension Reduction* UMAP) McInnes, Healy e Melville (2018) supera essas limitações, pois pode lidar facilmente com conjuntos de dados bastante grandes, preservando a estrutura local e global dos dados. A redução de dimensionalidade efetuada pelo UMAP é não linear e conhecida como aprendizado múltiplo. Ela emprega várias técnicas que visam projetar dados de alta dimensão em variedades latentes de dimensão inferior, com o objetivo de visualizar os dados no espaço de baixa dimensão ou aprender o mapeamento.

Projeção customizável (CUSTOM)

Finalmente, usando projeção customizável (*custom*) o usuário do *Embedding Projector* pode construir projeções lineares especializadas com base em pesquisas de texto para possibilitar obter "direções"significativas no espaço. Para isso o usuário deve inserir pelo menos duas strings de pesquisa ou expressões regulares. O programa calcula os centróides dos conjuntos de pontos cujos rótulos correspondem a essas buscas e utiliza o vetor de diferença entre os centróides como eixo de projeção.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma breve revisão bibliográfica de trabalhos que fazem extração de informação e mineração de textos com possibilidades de aplicação em análise de discursos literários. O processo adotado para a busca por trabalhos relacionados seguiu a metodologia de revisão sistemática proposta Kitchenham e Charters (2007). A pesquisa foi realizada entre janeiro de 2022 e julho de 2022 nos repositórios da *ACM Digital Library*, *IEEE Xplore Digital Library* e no *Google Scholar*. A expressão de busca utilizou os termos em inglês "embeddings", "pattern", "text", "morpho-semantic", "mining", "Processamento de Linguagem Natural", "BERT"e "model"e suas traduções para o português. A busca inicialmente envolveu publicações a partir de 2020. As publicações encontradas foram selecionadas de acordo com os seguintes critérios de inclusão: artigos completos publicados em periódicos ou conferências; teses defendidas; relacionados a área de computação; escritos em idioma inglês ou português; que abordam temas sobre padrões linguísticos e/ou análise de discurso. Após o processo de seleção e leitura dos resumos dos artigos, foram selecionadas quatro publicações com alta relevância para o tema deste trabalho. Elas são discutidos a seguir.

Goularte et al. (2020) propõe uma abordagem e métodos automatizados para induzir e desambiguar o sentido correto das palavras alvo em coleções de textos curtos. O trabalho analisa conteúdos gerados por usuários em redes sociais como *Twitter*, através da utilização de padrões léxico-semânticos *fuzzy* definidos como sequências de *Componentes Morfo-semânticos* (MSC). Tal abordagem utiliza técnicas de NER, NED, *Word Sense Induction* (WSI)/*Word Sense Disambiguation* (WSD) e *POS-Tagging*, para extrair dos textos informações que permitam identificar padrões e com eles realizar a desambiguação de palavras alvos que ferramentas existentes não conseguem desambiguar. A abordagem proposta tem 5 etapas. A primeira, o pré-processamento que realiza limpeza e normalização dos textos, além de uma filtragem para coletar somente os textos relevantes; A segunda etapa realiza a *POS-Tagging* (com uso da ferramenta *FreeLing*) para classificar cada token (palavra) do texto com seu respectivo lema e classe morfossintática; A terceira etapa determina os sentido de cada token, na medida das possibilidades utilizando NER/NED e WSI/WSD usando ferramentas existentes como a *FreeLing* e API para pesquisar *synset's* do *WordNet*. A quarta etapa realiza o casamento (*Matching*) de componentes morfo-semânticos (MSC) de textos distintos, segundo sua classificação morfológica e sentido, para encontrar padrões MSC^{+} recorrentes. Por último, se faz WSI e WSD com base nos padrõesMSC^{+}, para tokens com sentindo indefinido ou parcialmente desambiguados. Este trabalho definiu padrões MSC^{+}, forneceu um algoritmo (que não usa *embeddings*, mas casamento de classes POS-tags e de sentidos) para minerá-los e um método que usa tais padróes para desambiguar tokens com sentido difícil de capturar.

Sorato, Goularte e Fileto (2020) propuseram uma abordagem de mineração de padrões linguísticos centrados em uma palavra alvo (*Simple Semantic Patterns (SSP)* para apoiar análise de discursos. A mineração identifica fragmentos de textos semanticamente semelhantes que caracterizam os padrões SSP utilizando casamento por similaridade semântica de *embeddings*

concatenados de palavras adjacentes a uma palavra alvo. Os textos utilizados nos experimentos são de três conjuntos de dados rotulados: Waseem e Hovy (2016), Basile et al. (2019) e Davidson et al. (2017), referentes a discursos de ódio, sexistas, racistas, entre outros temas delicados, somando ao todo cerca de 40 mil *tweets* ofensivos ou não. A implementação usou o *NLTKTwitter tokenizer*, o *WordNetLemmatizer*, o *ekphrasis*, a biblioteca *scikit-learn*, *TF-IDF*, a biblioteca *FastText* e o modelo de *embedding* GloVe para realizar a mineração de padrões e a classificação dos discursos. Realizou-se uma tarefa de pré-processamento com a limpeza de pontuações, *emojis*, *URL* e menções de usuários, tokenização e lematização do texto. Posteriormente, fez-se uma filtragem para encontrar textos contendo ao menos uma palavra-alvo selecionadas: 275 palavras referentes a racismo e 131 referentes a sexismo. Em seguida, um algoritmo expande janelas ao seu redor de cada palavra alvo encontrada nos textos e compara os *embeddings* concatenados das palavras dentro de cada janela com os de outras janelas, usando similaridade do cosseno. A quantidade de palavras das janelas cresce até que a similaridade dos *embeddings* seja inferior a um *threshold*. As sequências de palavras similares encontradas são agrupadas segundo o grau de similaridade definido pelo *threshold*. Então os agrupamentos são analisados para identificar assuntos mencionados ao redor das palavras-alvo. Por fim, é possível realizar classificar se o texto tem ou não teor pejorativo utilizando como característica (*features*) de classificação os padrões *Short Semantic Pattern* (SSP), unigramas, bigramas, trigramas e sequências ponderadas TF-IDF de até 3 *Tags* de POS-*Tagging*. A técnica de mineração utilizando padrões SSP extrai declarações recorrentes dos conteúdos de *tweets* sem ter de realizar a revisão manualmente. Este trabalho fornece técnicas para minerar e analisar palavras-alvo em discurso utilizando janelas e *embeddings* de texto, mas não atuais *embeddings* contextualizados.

Araújo-Junior (2021) minera textos literários do tipo poema oriundos do Projeto Coletânea de Poesias, o qual coleta poesias de alunos em diversos níveis de ensino. O intuito é identificar os temas abordados pelos alunos, extrair características dos poemas e padrões contidos nos textos referentes às séries dos autores. A implementação da proposta usou o *nlpnet* para realizar *POS-Tagging*, o *Gibbs Sampling for the Dirichlet Multinomial Mixture* (GSDMM) para analisar os padrões, a biblioteca *scipy* para realizar a análise da correlação *tau* de *Kendall* e a *Normalized Pointwise Mutual Information* (NPMI) para verificar a associação entre duas palavras. O trabalho consistiu em realizar a recuperação dos poemas dos anos de 2010 até 2019 contabilizando, ao todo, 684 poemas das séries sexto ano do ensino fundamental até o terceiro grau. Realizou-se uma tarefa de pré-processamento, onde foram realizadas a limpeza e padronização dos textos. Também realizou-se a extração de características textuais gerais do conjunto de dados. Posteriormente foi executada a tarefa de POS-*Tagging* para mapear as classes gramaticais e palavras denotativas. Também foram realizadas algumas análises estatísticas das informações obtidas. A partir dos resultados levantados, foi realizada a análise descritiva dos dados, para validar se a série do aluno implica nas características levantadas anteriormente. Modelagem de tópicos foi usada para identificar grupos de termos que costumam aparecer juntos nos textos e identificar os tópicos e palavras que aparecem recorrentemente juntos nos textos. Os resultados obtidos através da mineração não possibilitam afirmar com precisão que as ca-

racterísticas identificadas não estão relacionadas com a série do aluno. Também não é possível distinguir um único perfil nas produções textuais devido à linguagem plurissignificativa que este gênero textual utiliza. Este trabalho fornece técnicas para mineração e análise de textos literários, porém não analisa discursos, mas somente tópicos e características gerais dos textos.

Braz-Junior e Fileto (2021) propuseram um modelo de coerência baseado no BERT que inclui um classificador binário e um mensurador de (in)coerência em documentos. O classificador utiliza o token de '[CLS]' para discriminar documentos originais de versões permutadas. O mensurador utiliza uma função de (in)coerência que compara embeddings de documentos utilizando medidas de similaridade cosseno e distâncias Euclidiana e Manhattan. Os embeddings dos documentos são consolidados utilizando estratégias de pooling MAX e MEAN. O modelo proposto teve uma acurácia de 97% na classificação de documentos. Este trabalho fornece técnicas para consolidar embeddings de documentos e mostra a viabilidade de se utilizar medidas de similaridade e distância na comparação dos embeddings gerados pelo BERT.

3.1 TABELA COMPARATIVA

A Tabela 4 fornece um resumo comparativo das propostas selecionadas e analisadas. Os trabalhos são ordenados cronologicamente nas linhas da tabela. Eles são comparados conforme os aspectos das soluções de análise de textos que consideramos mais relevantes, que aparecem nas colunas. A primeira coluna (**Autor e Ano**) define o nome dos autores do trabalho avaliado e o ano de publicação. A segunda coluna (**Área de Aplicação**) refere-se à origem dos textos avaliados. A terceira coluna (**Objetivos**) refere-se ao que se pretende alcança com os estudos propostos. A quarta coluna re refere ao tipo de ***embedding*** utilizado por cada trabalho. Na quinta coluna (**Abordagem**) indica as técnicas utilizadas. Por fim, a coluna **Ferramenta** referencia as bibliotecas/ ou *frameworks* utilizados.

Grande parte dos trabalhos analisados não usam *embeddings* contextualizados. Sorato, Goularte e Fileto (2020) utiliza *embeddings* estáticos como GloVe para minerar padrões em textos, enquanto Araújo-Junior (2021) e Goularte et al. (2020) não usam *embedding* nenhum e o Braz-Junior e Fileto (2021) utiliza *embeddings* contextualizados para classificador binário e um mensurador de (in)coerência em documentos. Nossa proposta pretende explorar representações de linguagem com base nos *embeddings* contextualizados produzidos pelo BERT, pois possuem representações que consideram o contexto em todas as direções. Além disso, o BERT permite ajuste-fino para tarefas *downstream*. Outra característica dos trabalhos relacionados é o fato deles manipularem textos curtos quando comparados aos textos literários considerados no nos presente trabalho, que pelo seu tamanho maior têm muito mais informação de contexto. Isso sugere que o uso do BERT, que é capaz de explorar esses contextos, pode contribuir na tarefa de encontrar padrões recorrentes envolvendo uma palavra-alvo em tais textos.

Tabela 4 – Quadro comparativo dos trabalhos relacionados

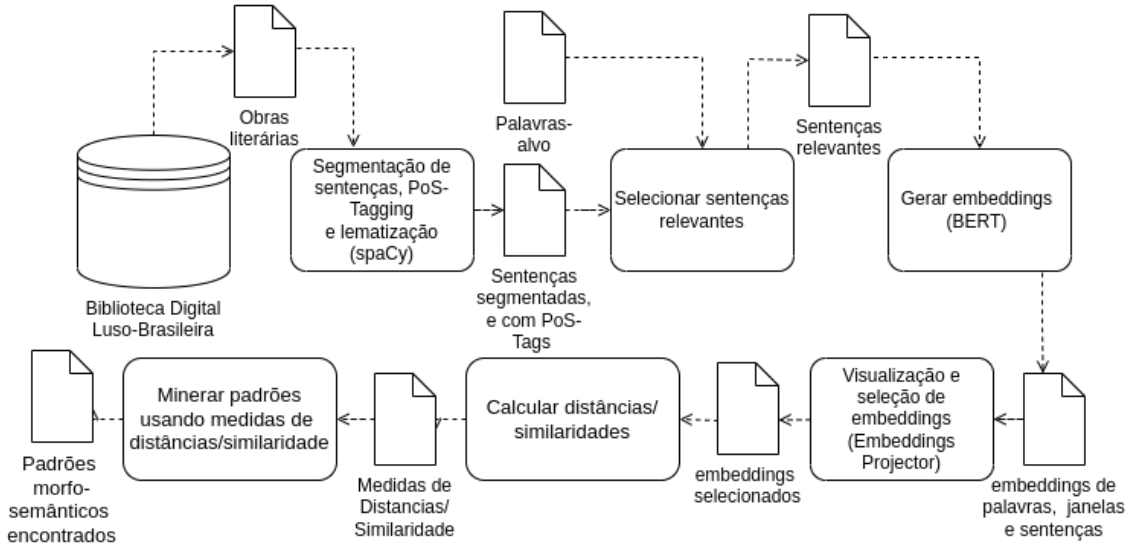
Trabalho	**Textos processados**	**Objetivos**	**Casamento semântico**	**Tarefas PLN**	**Ferramenta de PLN**
Sorato, Goularte e Fileto (2020)	tweets	Minerar padrões para analisar e classificar discursos	GloVe	POS-*Tagging*	spaCy, *scikit-learn*
Goularte et al. (2020)	postagens em mídias sociais	Minerar padrões para desambiguar palavras	classes de entidades reconhecidas	POS-*Tagging*, NER	*FreeLing*, *WordNet*
Araújo-Junior (2021)	Textos literários (Poemas)	Minerar Padrões	N/A	*POS-Tagging* correlação *tau* de *Kendall*, GSDMM, NPMI	*nlpnet*, *scipy*
Braz-Junior e Fileto (2021)	perguntas de QA	Analisar coerência de discursos	BERT	*POS-Tagging*	*nlpnet*, *scipy*, Google Colab
Nossa proposta	**Textos Literários**	**Minerar padrões para analisar e classificar discursos**	**BERTimbau**	*POS-Tagging*	**spaCy, Google Colab**

Fonte: o autor.

4 PROCESSO GERAL E ALGORITMO PARA MINERAR PADRÕES EM TEXTOS

Este capítulo descreve as etapas do processo proposto neste trabalho para minerar padrões morfo-semânticos em textos. A Figura 8 ilustra o fluxo de informação proposto para minerar os padrões em obras literárias da Biblioteca Digital Luso-Brasileira[1], embora o processo proposto possa ser aplicado a textos de diversas outras fontes. O processo se inicia com o uso do SpaCy para efetuar segmentação do texto em sentenças, POS-*Tagging* e lematização. Posteriormente, seleciona sentenças relevantes, isto é, que contenham palavras alvo, em torno das quais se deseja minerar padrões e analisar discursos. O *BERTimbau* é usado para gerar embeddings das palavras contidas nas sentenças selecionadas e o *Embedding Projector* para visualizar a distribuição desses *embeddings* e selecionar aqueles adequados para a mineração de padrões. Finalmente, os *embeddings* selecionados são comparados usando o algoritmo de Sorato, Goularte e Fileto (2020) e outros mais adequados a textos literários desenvolvidos no âmbito deste trabalho. Tais algoritmos avaliam a similaridade semântica das palavras usadas nos discursos em torno das palavras alvo para identificar os padrões semânticos, isto é instâncias de texto com os mesmo sentidos em torno das mesmas palavras alvo, embora muitas vezes usando construções léxicas e sintáticas distintas.

Figura 8 – Mineração de padrões morfo-semânticos em textos.



Fonte: o autor.

[1] https://www.literaturabrasileira.ufsc.br/

4.1 SEGMENTAÇÃO DO TEXTO EM SENTENÇAS, *POS-TAGGING* E LEMATIZAÇÃO

A primeira etapa do processo proposto é o pre-processamento dos textos. Nesta etapa primeiro se faz a padronização dos textos, mediante remoção de certos caracteres especiais (e.g.,$\backslash n$,$\backslash xad$) utilizados nas pagina da internet, remoção das repetições de espaços em branco, pontuações e símbolos (e.g., ponto final (.), ponto interrogação (?), ponto exclamação (!), traço ($-$)). Os textos padronizados são então submetidos à biblioteca *spaCy* para realizar as tarefas de tokenização, segmentação dos textos em sentenças, lematização e POS-*Tagging*. Após o término desses procedimentos, dados são armazenados utilizando uma estrutura semi-estruturadada, contendo id de cada sentença, lista de seus tokens, lista de lemmas, lista de POS-*Tags*, lista de verbos e a própria sentença. Esses dados são utilizadas para facilitar a seleção das sentenças relevantes para mineração de padrões.

4.2 SELECIONAR SENTENÇAS RELEVANTES

A tarefa de selecionar sentenças relevantes é realizada utilizando os dados gerados na etapa anterior (descrita na Seção 4.1). Uma sentença é selecionada se contém ao menos uma das palavras-alvo fornecidas pelos especialistas do domínio. O casamento desconsidera maiúsculas e minúscula (i.e., não é *case-sensitive*). As palavras-alvo estão separadas em grupos contendo as flexões consideradas de cada uma:

- **branco**: branca, brancas, branco, brancos
- **criado**: criada, criadas, criadinha, criadinhas, criadinho, criadinhos, criado, criados
- **crioulo**: crioula, crioulas, crioulinha, crioulinhas, crioulinho, crioulinhos, crioulo, crioulos
- **escravo**: escrava, escravas, escravinha, escravinhas, escravinho, escravinhos, escravo, escravos
- **mulato**: mulata, mulatas, mulatinha, mulatinhas, mulatinho, mulatinhos, mulato, mulatos
- **negro**: negra, negras, negrinha, negrinhas, negrinho, negrinhos, negro, negros
- **senhor**: senhor, senhora, senhoras, senhores

Como uma sentença pode conter mais do que uma palavra-alvo é mantida a posição de cada ocorrência. Isso é importante para identificar corretamente as instâncias candidatas de padrões morfo-semânticos.

4.3 GERAR *EMBEDDINGS*

Nesta etapa, as sentenças relevantes são submetidas ao BERTimbau pré-treinado na língua portuguesa na sua versão grande ($BERTimbau_{Large}$) para gerar os *embeddings*. O texto de entrada (*input*) do BERTimbau pré-treinado é limitado a 512 *tokens*. Assim, sentenças relevantes são submetidas individualmente ao BERTimbau para não extrapolar este limite enquanto

se mantém a informação de contexto de cada sentença. São gerados *embeddings* de componentes textuais em 3 níveis de granularidade: sentenças, janelas dentro de sentenças e palavras individuais. O *embedding* de cada sentença é a média dos *embeddings* de seus *tokens*. Janelas são fragmentos de sentenças centrados em uma palavra-alvo, considerando um certo número (usualmente 1 a 10) de palavras vizinhas à esquerda e à direita, até no máximo o limite da sentença. O *embedding* de uma janela é a concatenação dos *embeddings* dos *tokens* que estão dentro dela. Esses *embeddings* de *tokens* são coletados dos *embeddings* da respectiva sentença. Isso é feito por dois motivos: para obter *embeddings* de janelas que considerem todo o contexto da sentença e para capturar relações da sentença com a janela. Por fim, os *embeddings* de palavras são os *embeddings* dos respectivos *tokens* da sentença. Os *embeddings* de *tokens* são a última camada do BERT, gerando embeddings com 1024 valores (dimensões).

Algoritmo 1: Criar lista de janelas

Data: Lista de sentenças("*idSentenca*","*tokens*", "*postagging*", "*palavraAlvo*", "*grupo*", "*posAlvo*") de sentenças relevantes (*SR*)
Result: Lista de janelas(registro)

```
1.  listaRegistro ← [ ]
2.  foreach index, reg in SR do
3.      idSentenca ← reg["idSentenca"]
4.      tokens ← reg["tokens"]
5.      posAlvo ← reg["posAlvo"]
6.      palavraAlvo ← reg["palavraAlvo"]
7.      grupo ← reg["grupo"]
8.      texto ← juncao(tokens)
9.      embSentenca ← getEmbeddingsText(texto)
10.     tamJanela ← 1
11.     janelaInf, janelaSup, expande ← expandeJanela(token, posAlvo, posAlvo)
12.     while expande = True do
13.         embJanela ← embSentenca[janelaInf : janelaSup]
14.         janela ← juncao(tokens[janelaInf : janelaSup])
15.         compJanela ← tamJanela * 2 + 1
16.         registro ← [idSentenca, texto, janela, compJanela,
              embJanela, palavraAlvo, grupo]
17.         listaRegistro.append(registro)
18.         janelaInf, janelaSup, expande ←
              expandeJanela(token, janelaInf, janelaSup)
19.         tamJanela ← tamJanela + 1
```

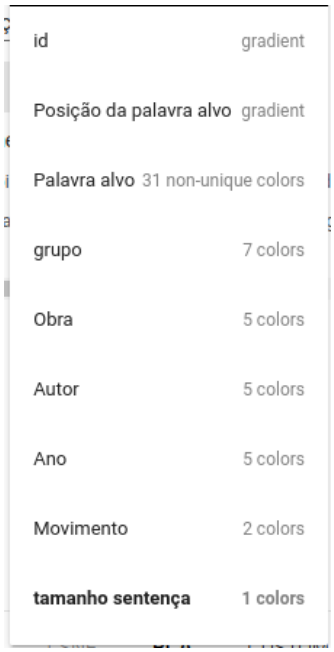
O Algoritmo 1 apresenta na forma de pseudocódigo o programa criado para a tarefa de gerar *embeddings* de sequências de palavras (janelas) que contenham uma determinada palavra-alvo. As sentenças utilizadas nessa tarefa são selecionadas como descrito na Seção 4.2. Uma lista de registros é criada para armazenar as janelas com seus *embeddings*. Todas as sentenças relevantes são percorridas para gerar os *embeddings*. O registro de cada sentença da lista é composto pelo id da sentença, lista de *tokens*, posição da palavra alvo, a própria palavra

alvo e grupo da palavra-alvo. Para cada sentença é realizada a concatenação dos *tokens* utilizando a função *"juncao"*. O texto resultante da junção é passado como parâmetro para a função "*getEmbeddingsText*", para gerar e retornar os seus *embeddings*. Em seguida, a função "*"expandeJanela"* recebe o *token* e sua posição para gerar e retornar os limites inferiores e superiores da janela. Com os limites definidos, é criado o registro da janela contendo: o id da sentença, a sentença, o texto da janela, o comprimento da janela, os *embeddings* de tokens dentro da janela, a palavra-alvo e o grupo desta palavra. O registro é então adicionado à lista de registros. Por fim, é realizada a expansão da janela atual, passando os *tokens* da sentença e os limites da janela acrescentando um *token* de cada lado, se o limite da sentença permitir. Esse processo de expansão ocorre até que não seja mais possível expandir o tamanho da janela dentro da sentença.

4.4 VISUALIZAÇÃO E SELEÇÃO DE EMBEDDINGS

Nesta etapa, os dados gerados como explicado na seção 4.3 são utilizados para gerar dois arquivos *.tsv* padronizados que servem de entrada para o *Embedding Projector*. Um desses arquivos contém a lista de *embeddings* a visualizar. O outro arquivo contém rótulos (*labels*) para representar características dos respectivos *embeddings* (e.g., obra, autor, ano, movimento literário, classe morfossintática ou classe de sentido de *embedding* de palavra, palavra-alvo de um *embedding* de janela). O *Embedding Projector* escolhe uma cor de uma lista de cores para representar *embeddings* rótulos distintos de uma característica (e.g. autores distintos). Isso possibilita identificar uma característica escolhida de cada ponto na visualização (*embedding)* de acordo com a cor usada para exibí-lo.

Figura 9 – Label de característica do *Embeddings Projector*



Fonte: o autor.

A Figura 9 apresenta uma lista de características dos *embeddings* de janelas com tamanho 7 que podem ser exibidas, com os respectivos números de cores à direita. A indicação *gradient* no lugar do número de cores corresponde ao uso de degradê, devido ao alto número de rótulos possíveis para a respectiva característica.

O *Embedding Projector* possibilita avaliar a distribuição espacial e a vizinhanças dos *embeddings* nele carregados. Isso pode ser feito usando uma das técnicas apresentadas na Subseção 2.3.1 para verificar se os *embeddings* quando projetados em duas ou três dimensões apresentam grupos bem definidos de pontos, polarização ou não servem para o experimento por terem distribuição muito esparsa, sem formar grupos. Essa visualização dos *embeddings* projetos em duas ou três dimensões pode ser feita por especialistas do domínio. Se o conjunto de dados apresentar alguma propriedade relevante ou desejável, ele pode ser selecionado e separado para ser utilizado na tarefa na mineração de padrões.

4.5 CALCULAR DISTÂNCIAS/SIMILARIDADES

O cálculo das medidas de similaridade e distância é realizado para cada corpus e respectivos *embeddings* selecionados pelos especialistas de domínio. São calculadas a distância (Euclidiana e Manhathan) e a similaridade (cosseno) para todos os pares de *embeddings* de palavras, de janelas e de sentenças. Todas as distâncias entre pares de *embeddings* ficam pré-calculadas e armazenadas para reuso pela próxima e última etapa do processo proposto, a mineração de padrões, a fim de evitar recalculá-las para os mesmos em execuções de algoritmos de mineração de padrões.

O Algoritmo 2 apresenta na forma de pseudocódigo o procedimento criado para realizar a tarefa de calcular distâncias/similaridades entre janelas que contenham uma palavra-alvo. Uma lista de registros é criada para armazenar as medidas entres janelas. Todos os *embeddings* de janelas são percorridas e comparados entre si. O registro de cada *embeddings* de janelas da lista é composto pelo id da sentença ("idSentenca"), sentença, janela, tamanho da janela ("compJanela"), lista de *embeddings* ("embJanela"), palavra-alvo e grupo da palavra-alvo. Para calcular as medidas de distâncias/similaridades é realizada a comparação de todos os *embeddings* de janelas ($n^2 - n$). A comparação dos *embeddings* das janelas é realizada pela função "*getMeasurementsEmbedding*"que recebe os *embeddings* de duas janelas para gerar e retornar as medidas de comparação. Com as medidas geradas é criado o registro $regcomp_a$ contendo os dados da comparação da janela a com b e o registro $regcomp_b$ da comparação da janela b com a. Cada registro é formado por: id da sentença, sentença, janela, tamanho da janela, palavra-alvo, grupo da palavra-alvo, id da sentença comparada, sentença comparada, tamanho da janela comparada, palavra-alvo comparada, grupo da palavra-alvo comparada e a medida. Por fim, os registros são inseridos na lista de medidas de janelas.

Algoritmo 2: Criar lista de medidas entre janelas

Data: Lista de embeddings de janelas ("*idSentenca*", "*sentenca*", "*janela*", "*compJanela*", "*embJanela*", "*palavraAlvo*", "*grupo*") de sentenças relevantes (*SR*)
Result: Lista de registro

```
1.  listaRegistro ← [ ]
2.  foreach index_a, reg_a in SR do
3.      idSentenca_a ← reg_a["idSentenca"]
4.      sentenca_a ← reg_a["sentenca"]
5.      janela_a ← reg_a["janela"]
6.      compJanela_a ← reg_a["compJanela"]
7.      embJanela_a ← reg_a["embJanela"]
8.      palavraAlvo_a ← reg_a["palavraAlvo"]
9.      grupo_a ← reg_a["grupo"]
10.     foreach index_b, reg_b in SR do
11.         idSentenca_b ← reg_b["idSentenca"]
12.         sentenca_b ← reg_b["sentenca"]
13.         janela_b ← reg_b["janela"]
14.         compJanela_b ← reg_b["compJanela"]
15.         embJanela_b ← reg_b["embJanela"]
16.         palavraAlvo_b ← reg_b["palavraAlvo"]
17.         grupo_b ← reg_b["grupo"]
18.         if idSentenca_a != idSentenca_b OR janela_a != janela_b then
19.             medida ← getMeasurementsEmbedding(embJanela_a, embJanela_b)
20.             regcomp_a ←
                  [idSentenca_a, sentenca_a, janela_a, compJanela_a, palavraAlvo_a, grupo_a,
                   idSentenca_b, sentenca_b, janela_b, compJanela_b, palavraAlvo_b, grupo_b,
                   medida]
21.             regcomp_b ←
                  [idSentenca_b, sentenca_b, janela_b, compJanela_b, palavraAlvo_b, grupo_b,
                   idSentenca_a, sentenca_a, janela_a, compJanela_a, palavraAlvo_a, grupo_a,
                   medida]
22.             listaRegistro.append(regcomp_a)
23.             listaRegistro.append(regcomp_b)
```

4.6 MINERAÇÃO DE PADRÕES MORFO-SEMÂNTICOS

Finalmente, a tarefa de minerar padrões morfo-semânticos em torno de palavras-alvo nas sentenças dos textos pode usar qualquer uma das medidas de distância ou similaridade entre *embeddings* (de palavras, de janelas de texto dentro de sentenças ou de sentenças inteiras) calculadas na etapa anterior do processo proposto (Seção 4.5). O Algoritmo 3 descreve na forma de pseudocódigo a função criada para minerar padrões morfo-semânticos com base em alguma medida de similaridade entre *embeddings* de janelas de texto dentro de sentenças. Um dicionário é criado para armazenar as ocorrências de janelas. Todas as janelas são percorridas para realizar o agrupamento delimitado pelo valor do *threshold*. Cada registro é formado pelo

id da sentença, sentença, janela, comprimento da janela, palavra-alvo, grupos de cada janelas comparada e a medida. Para cada registro é verificado se a medida está acima de um *threshold*. A medida estando acima do *threshold* é gerado um identificador formado pela concatenação do id da sentença, a palavra-alvo e janela. O identificador será a chave única no dicionário. Para cada novo identificador é criado um conjunto vazio no dicionário. Para cada ocorrência de um identificador é adicionado o registro ao conjunto de sua respectiva chave. Finalmente é possível realizar uma contagem de cada item desse dicionário e identificar qual janela apresenta a maior ocorrência.

Algoritmo 3: Mineração de padrões morfo-semânticos usando *embeddings* de janelas textuais

Data: Lista de medidas de janelas("$idSentenca_a$", "$sentenca_a$", "$janela_a$", "$compJanela_a$", "$palavraAlvo_a$", "$grupo_a$", "$idSentenca_b$", "$sentenca_b$", "$janela_b$", "$compJanela_b$", "$palavraAlvo_b$", "$grupo_b$", "$medida$")

Result: dicionario de ocorrências de padrões semânticos em janelas

```
1.  dicionario ← dict()
2.  foreach index, reg in listaJanelas do
3.      idSentenca ← reg["idSentenca_a"]
4.      palavraAlvo ← reg["palavraAlvo_a"]
5.      janela ← reg["janela_a"]
6.      medida ← reg["medida"]
7.      if medida >= threshold then
8.          identificador = idSentenca + "__" + palavraAlvo + "__" + janela
9.          if (identificador in dicionario) == False then
10.             dicionario[identificador] ← set()
11.         dicionario[identificador].append(reg)
12. foreach index, janela in dicionario do
13.     if (janela in dicionario) == False then
14.         dicionario[janela] ← 1
15.     dicionario[janela] ← dicionario[janela] + 1
```

5 EXPERIMENTOS E RESULTADOS

Esta seção relata os experimentos para avaliar a nossa proposta para minerar padrões morfo-semânticos em textos literários visando apoiar análise de discursos. Primeiramente, a Seção 5.1 descreve alguns detalhes da implementação do processo proposto. A Seção 5.2 descreve os conjuntos de dados usados nos experimentos. A Seção 5.3 analisa algumas características gerais dos textos das obras literárias utilizada nos experimentos. A Seção 5.4 apresenta as análises e visualizações dos *embeddings* obtidos do conjunto de dados da obra. Por fim, a Seção 5.6 apresenta e discute os resultados obtidos pela mineração de padrões morfo-semânticos.

5.1 IMPLEMENTAÇÃO

O processo proposto foi implementado na linguagem de programação *Python* versão 3.7.13, sobre notebooks do ambiente de execução Google colaboratory [1]. O uso de notebooks visa facilitar a implementação, demonstração e a avaliação dos resultados obtidos. O ambiente Colaboratory também viabiliza experimentos que requerem computadores de alto desempenho com unidades de processamento gráfico (do inglês, *Graphics Processing Unit* - GPU) e unidades de processamento de tensores (do inglês, *Tensor Processing Unit* - TPU) para serem realizados em tempo hábil. A linguagem Python do ambiente vem pré-configurada facilitando o uso da biblioteca Transformers (WOLF et al., 2020) versão 4.5.1 da Huggingface [2], um provedor de código aberto de tecnologias de PLN que implementa a arquitetura padrão do BERT. Dentre os modelos do BERT pré-treinados, utilizamos um modelo treinado para a língua portuguesa *BERTimbau*[3] (SOUZA; NOGUEIRA; LOTUFO, 2020) no tamanho *large*, no formato "*cased*" (com caracteres maiúsculos e minúsculos) disponíveis gratuitamente. Além da ferramenta spaCy [4] versão 3.2.0, com vários recursos para processamento de linguagem natural, incluindo funcionalidades para segmentação dos documentos e a análise sintática das sentenças.

5.2 CONJUNTO DE DADOS

Utilizamos um conjunto de dados sugerido por especialistas em literatura e linguística, composto por 5 obras literárias de autores brasileiros com temas relacionados escravidão, abolicionismo e questões sociais. Os textos integrais dessas obras estão disponíveis na BLPL [5]. A Tabela 5 apresenta os nomes das obras em ordem alfabética (coluna "Nome"), nome do autor ("Autor"), Ano de Publicação, Movimento Literário e Temática. Note que essas obras foram publicas entre 1869 e 1890, sendo na maioria pertencentes ao movimento literário romantismo e uma delas pertencente ao realismo.

[1] https://colab.research.google.com/notebooks/intro.ipynb
[2] https://huggingface.co/transformers/index.html
[3] https://github.com/neuralmind-ai/portuguese-bert/
[4] https://spacy.io/
[5] https://www.literaturabrasileira.ufsc.br/?locale=pt_BR

Tabela 5 – Conjunto das obras literárias

Nome	**Autor**	**Ano de Publicação**	**Movimento Literário**	**Temática**
A Escrava Isaura	Bernardo Guimarães	1875	Romantismo	Abolicionista
As Vítimas-Algozes	Joaquim Manuel de Macedo	1869	Romantismo	Escravidão
Memórias Póstumas de Brás Cubas	Machado de Assis	1880	Romantismo	Retrata a escravidão, as classes sociais, o cientificismo e o positivismo da época
O Cortiço	Aluísio Azevedo	1890	Realismo	Trata das questões sociais e outra que trata das questões individuais e sentimentais
Úrsula	Maria Firmina	1859	Romantismo	Negritude a partir da perspectiva do próprio negro

Fonte: o autor.

A Tabela 6 apresenta estatísticas sobre a quantidade de sentenças, palavras e tokens encontrado nos textos das obras literárias consideradas, após as tarefas de segmentação de sentenças e geração de *emmbeddings*. Para cada obra é apresentado da quantidade de sentenças ("Qtd. Sentença"), quantidade de palavras por sentença ("Qtd. Palavras"), quantidade de tokens gerados pelo BERT ("Qtd. tokens BERT ") e quantidade e percentual de palavras desconhecidas pelo BERT ("Qtd.e % de Palavras desconhecidas"), para as quais o BERT gera ngrams (subpalavras com *n* caracteres).

Tabela 6 – Quantidades de sentenças, palavras e tokens nas obras consideradas

Nome	**Qtd. Sentenças**	**Qtd. Palavras**	**Qtde. tokens BERT**	**Qtde. e % Palavras desconhecidas**
A Escrava Isaura	3.581	6.6348	86.143	5.871 (8,85%)
As Vítimas-Algozes	6.044	121.192	160.451	8.715 (7,19%)
Memórias Póstumas de Brás Cubas	4.678	78.371	103.617	7.220(9,21%)
O Cortiço	6.163	101.582	132,889	8.442 (8,31%)
Úrsula	3.439	56.241	75.574	5.090 (9,05%)

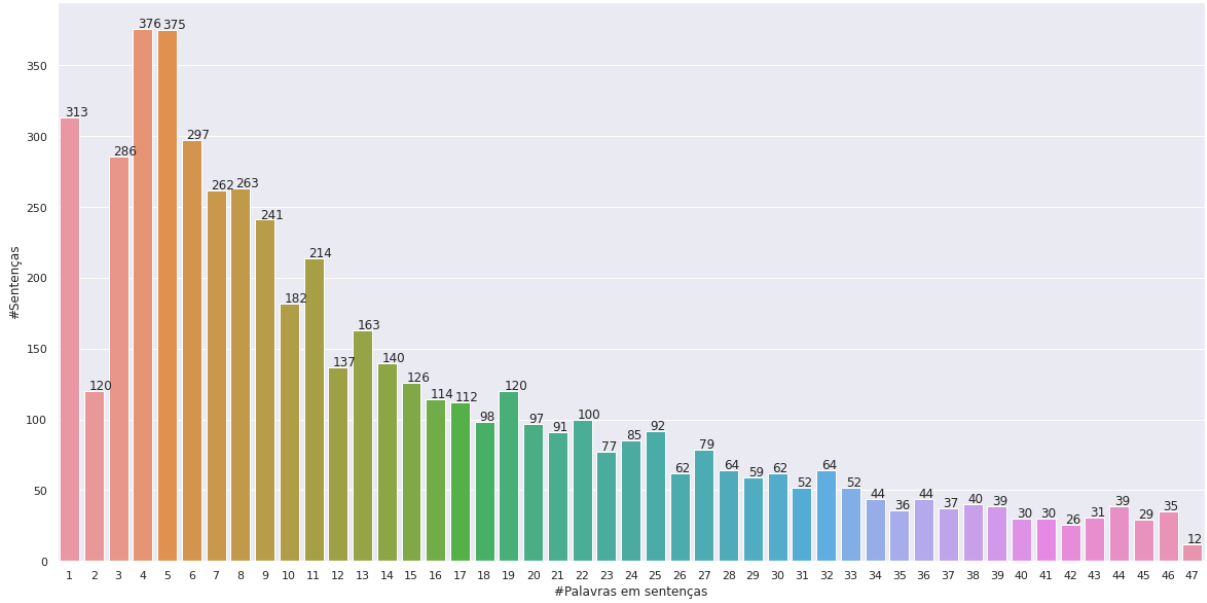
Fonte: o autor.

5.3 CARACTERIZAÇÃO DOS TEXTOS

Uma caracterização mais detalhada do texto foi realizada para as 5 obras. A ferramenta *spaCy* foi usada para realizar a segmentação do texto em sentenças, *POS-Tagging* e lematização. Após a segmentação e geração dos *embeddings*, foi verificado que a obra *As Vítimas-Algozes* apresenta um menor percentual de tokens desconhecidos pelo BERT que tende a minimiza a perda de informação no processo de *pooling MEAN* para gerar os *embeddings*. Além disso, tal obra também contem a maior quantidade de palavras dentre as obras analisadas. Por esses motivos utilizaremos esta obra na apresentação dos resultados das tarefas iniciais do processo proposto neste trabalho.

A Figura 10 apresenta a quantidade de sentenças com quantidade de palavras de 1 até 47 da obra *As Vítimas-Algozes*. Foram encontradas sentenças com até 247 palavras nesta obra, mas nesta figura excluímos quantidades acima de 47 para facilitar a visualização da distribuição. A obra apresenta 25% de suas sentenças com até 26 palavras.

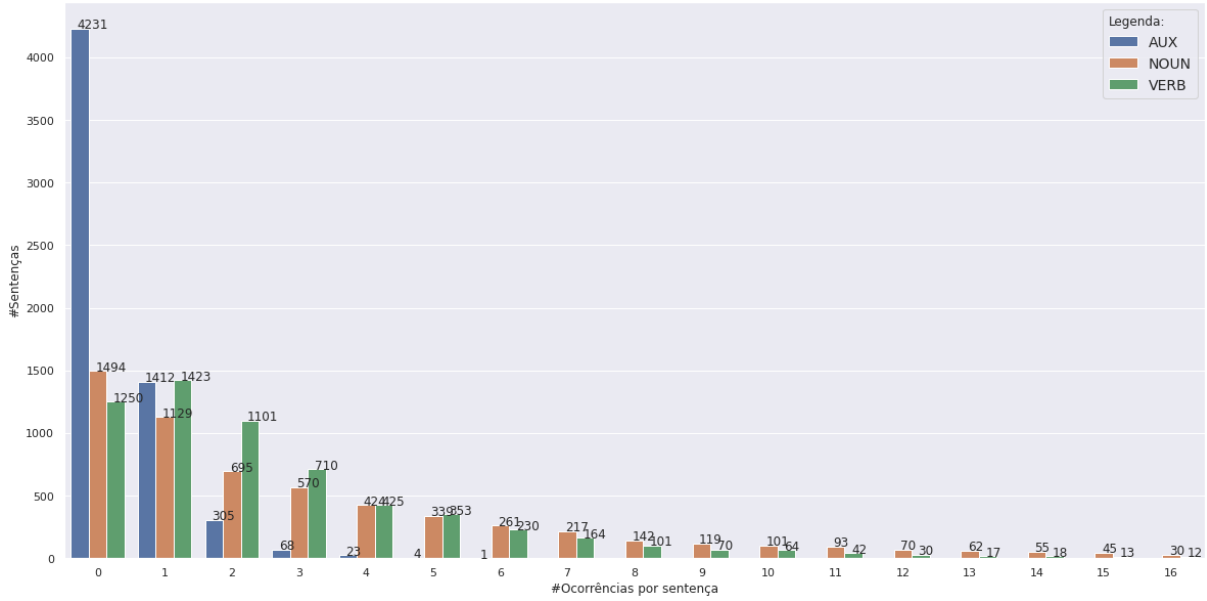
Figura 10 – Quantidade de sentenças por quantidade de palavras



Fonte: o autor.

A Figura 11 apresenta as quantidades de palavras encontradas nas sentenças da obra *As Vítimas-Algozes* com as classes morfossintáticas verbo ("VERB"), substantivo ("NOUN") verbo auxiliar ("AUX). Pode-se observar que a maioria das sentenças não tem verbo auxiliar, tem entre 1 e 3 verbos e uma grande quantidade de sentenças sem substantivo (i.e. 1.494).

Figura 11 – Quantidade de verbos, verbos auxiliares e substantivo por quantidade de palavras



Fonte: o autor.

5.4 ANÁLISE E VISUALIZAÇÃO DOS EMBEDDINGS

A distribuição das distâncias entre *embeddings* de palavras, janelas e sentenças foi analisada para avaliar a viabilidade do seu uso na mineração de padrões morfo-semânticos. A título de ilustração, a seguir é apresentada a análise das distâncias entre janelas textuais em torno da palavras-alvo **escravo**, fornecida pelos especialistas de domínio, em duas sentenças escolhidas aleatoriamente:

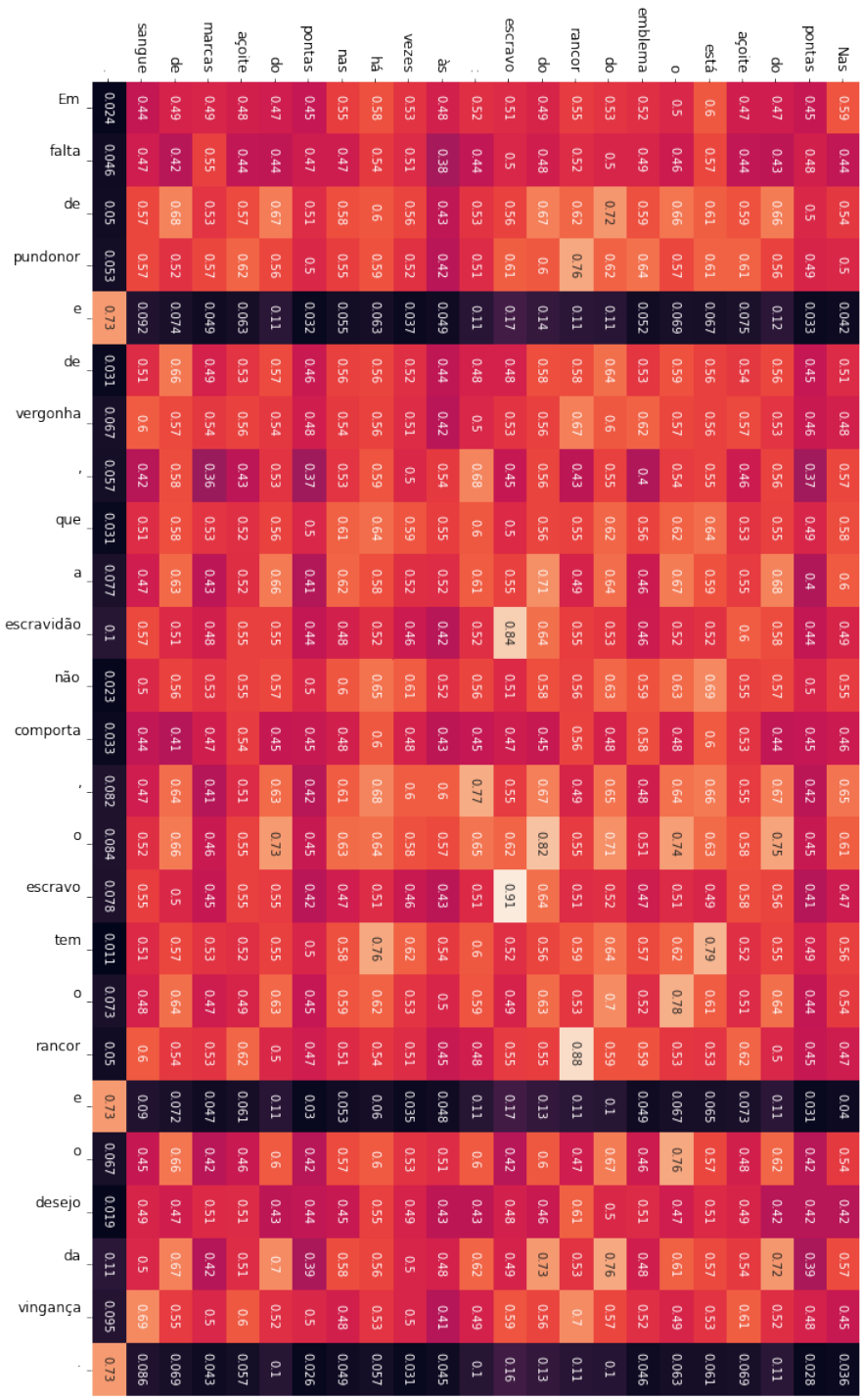
1. *"Em falta de pundonor e de vergonha, que a escravidão não comporta, o* ***escravo*** *tem o rancor e o desejo da vingança."*

2. *"Nas pontas do açoite está o emblema do rancor do* ***escravo****: às vezes há nas pontas do açoite marcas de sangue."*

As sentenças selecionadas foram submetidas ao BERT para gerar os *embeddings* contextualizados de suas palavras. A Figura 12 apresenta um mapa de calor (do inglês, *Heatmap*), que ilustra a similaridade utilizando a função do cosseno entre os *embeddings* de palavras de duas sentenças distintas da obra *As Vítimas-Algozes*. O mapa foi colocado na vertical para legibilidade das palavras das sentenças. Neste mapa, cada célula do mapa possui uma graduação em cores indicando o calor, ou seja, quanto mais diferente (próximo 0) forem as palavras, mais frio (escuro) e quanto mais similar (próximo de 1) mais quente (claro).

No mapa de calor da Figura 12 é possível identificar que os *embeddings* contextualizados da palavras-alvo "escravo" nas duas sentenças são distintos (devido aos diferentes contextos

Figura 12 – Mapa de calor entre os *embeddings* das palavras de duas sentenças da obra *As Vítimas-Algozes*.



Fonte: o autor.

textuais) mas bastante similares entre si. Também verifica-se que a similaridade da palavra "escravo"da sentença 2 (eixo y) com a palavra "escravidão"da sentença 1 (eixo x) é maior que $0,8$ e que as similaridades entre "rancor"e "pundonor" e entre "rancor" e "vingança"são maiores ou iguais a $0,7$. Esta análise inicial permitiu definir alguns patamares para similaridades além de fornecer indícios que existe certa similaridade entre os *embeddings* das palavras-alvo e de

outras palavras de cada sentença.

Após comparar os *embeddings* de palavras das duas sentenças, em uma matriz análoga à do mapa de calor da Figura 12, comparamos os *embeddings* dos tokens da sentença 1 entre si. A Figura 13 apresenta ao cento a palavra alvo "escravo"(na posição 16 da sentença 1) e os tokens adjacentes à sua esquerda e à sua direita na sentença 1. Para cada token é mostrada a sua posição ("W_i") na sentença, a similaridade do cosseno ("cos") do seu *embeddings* com a da palavra-alvo (escravo), sua classe morfossintática e o próprio token. Note que os tokens de cada lado da palavra-alvo na sentença estão ordenados nas respectivas tabelas (à direita e à esquerda na figura) segundo a similaridade cossenos dos seus *embeddings* com o da palavra-alvo. Palavras com a mesma classe morfossintática da palavra-alvo são destacadas em vermelho.

Figura 13 – Similaridade cosseno de tokens adjacentes à palavra "escravo"na sentença 1.

W_i	cos	classe	Palavra
11	0.8480	NOUN	escravidão
15	0.6360	DET	o
10	0.5526	ADP	a
14	0.5498	PUNCT	,
4	0.5000	NOUN	pundonor
7	0.4896	NOUN	vergonha
3	0.4889	ADP	de
12	0.4521	ADV	não
9	0.4389	SCONJ	que
1	0.4352	ADP	Em
6	0.4346	ADP	de
8	0.4305	PUNCT	,
2	0.4087	NOUN	falta
13	0.3798	VERB	comporta
5	0.1458	CCONJ	e

Adjacentes

"Escravo" 16 NOUN

Adjacentes

Palavra	classe	cos	W_i
vingança	NOUN	0.5386	25
rancor	NOUN	0.5252	19
da	ADP	0.4991	23
tem	VERB	0.4894	17
o	DET	0.4874	18
desejo	NOUN	0.4655	22
o	DET	0.4424	21
e	CCONJ	0.1443	20
.	PUNCT	0.1413	25

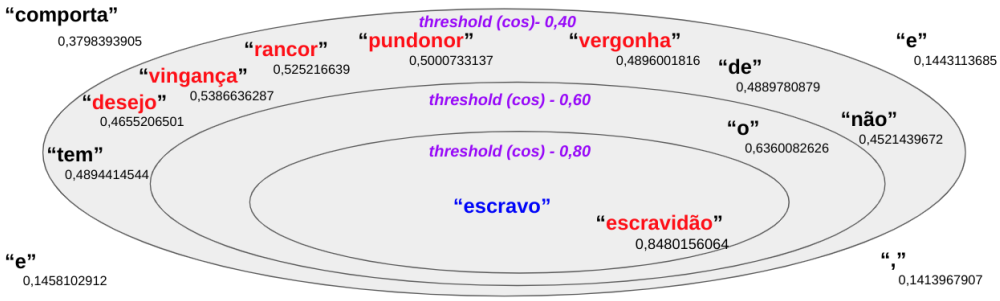
Fonte: o autor.

A Figura 14 ilustra a proximidade dos *embeddings* das palavras da sentença 1 com o da palavra-alvo, comem elipses concêntricas delimitando patamares de similaridades (i.e., *thresholds* $0,8$, $0,6$ e $0,4$). Percebe-se que vizinhança das palavras na sentença não tem relação com a similaridade do cosseno de seus respectivos *embeddings* com os da palavra alvo. Por exemplo, a palavra "escravidão"tem similaridade maior que 0,8 com a palavra-alvo nos *embeddings*, mas esta se encontra 5 palavras à esquerda da palavra-alvo "escravo". Além disso, há palavras de mesma classe morfossintática (substantivo) com *embeddings* semelhantes (similaridade maior do que $0,4$) ao da palavra-alvo (i.e. "desejo", "vingança", "rancor", "pundonor"), assim como algumas palavras de outras classe (i.e. "tem", "de", "não").

A Figura 15 complementa a análise das similaridades entre *embeddings* projetando em três dimensões os *embeddings* das palavras da sentença 1 e 2, com o uso da ferramenta *Embedding Projector*, a qual tem código disponível no github[6]. Nesta figura, é possível verificar que os *embeddings* da palavra-alvo *"escravo"* nas duas sentenças ficam próximos aos de palavras lexicamente similares como *"escravidão"*. Porém, palavras que não compartilham o

[6] https://github.com/dev-leandrodias/BERTimbautv1visualizacao

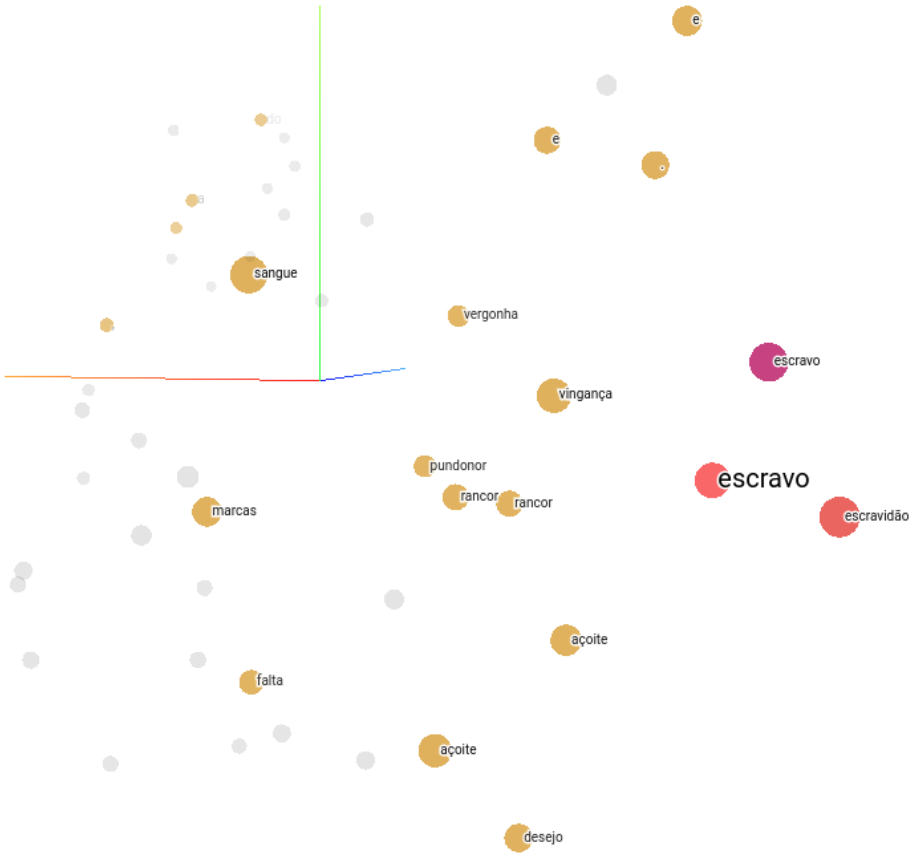
Figura 14 – Níveis de similaridade pelo *threshold*



Fonte: o autor.

mesmo léxico nem o mesmo sentido também ficaram próximas, tais como: *"açoite"*, *"pundonor"*, *"rancor"*, *"vingança"*. Assim, podemos dizer que essas palavras estão relacionadas, visto que, geralmente, são utilizadas juntas para compor informação nos mesmos contextos textuais.

Figura 15 – Projeção 3D de *embeddings* de palavras próximas aos da palavra-alvo "escravo".



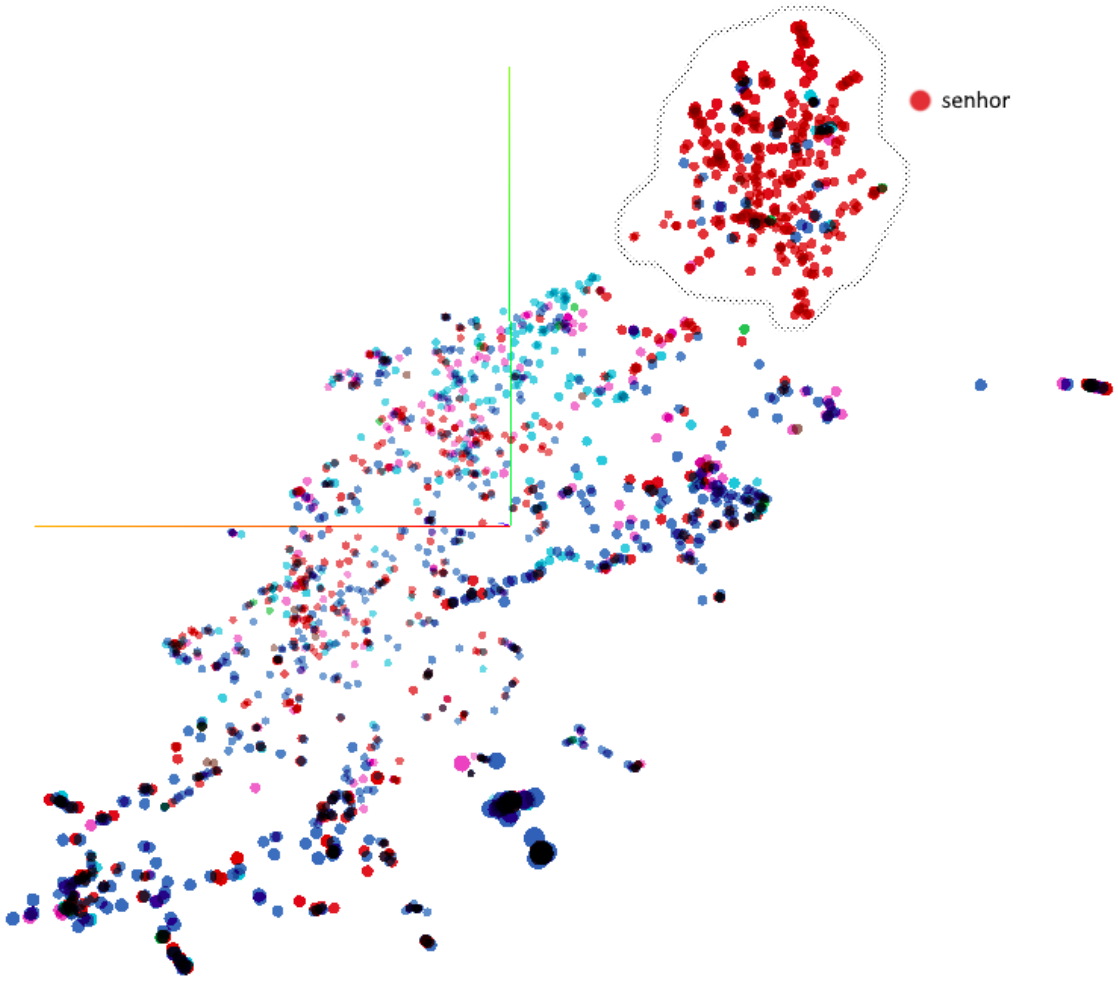
Fonte: o autor.

Considerando aos resultados promissores nos experimentos com as duas sentenças, realizamos o processamento de todas as sentenças da obra *As Vítimas-Algozes*. Realizamos a segmentação obtendo 6.044 sentenças, posteriormente selecionamos as sentenças relevantes

que possuem pelo menos uma palavra alvo, resultando em 1.844 sentenças. Com estas sentenças obtivemos 3 listas: lista dos *embeddings* consolidados das palavras de cada sentença, lista dos *embeddings* consolidados das janelas de tamanho 3 nas sentenças e a lista dos *embeddings* das palavras de cada sentença.

A Figura 16 apresenta a visualização em três dimensões utilizando projeção UMAP com 20 vizinhos (*neighbors*) dos *embeddings* das sentenças consolidados com palavras relevantes da obra *As Vítimas-Algozes*. Nesta figura, cada cor identifica sentenças com uma determinada palavra-alvo. Percebe-se um agrupamento de pontos em vermelho (destacado) no canto superior direito, referente a sentenças que possuem palavras-alvo do grupo "senhor". Os grupos de outras palavras-alvo estão em outras cores: marrom para o grupo da palavra-alvo "branco", verde para o grupo da palavra-alvo "criado", azul-claro para o grupo de "crioulo", azul para o grupo o grupo de "escravo"e cor rosa para o grupo de "negro".

Figura 16 – Projeção UMAP 3D de *embeddings* consolidados de sentenças da obra *As Vítimas-Algozes*.

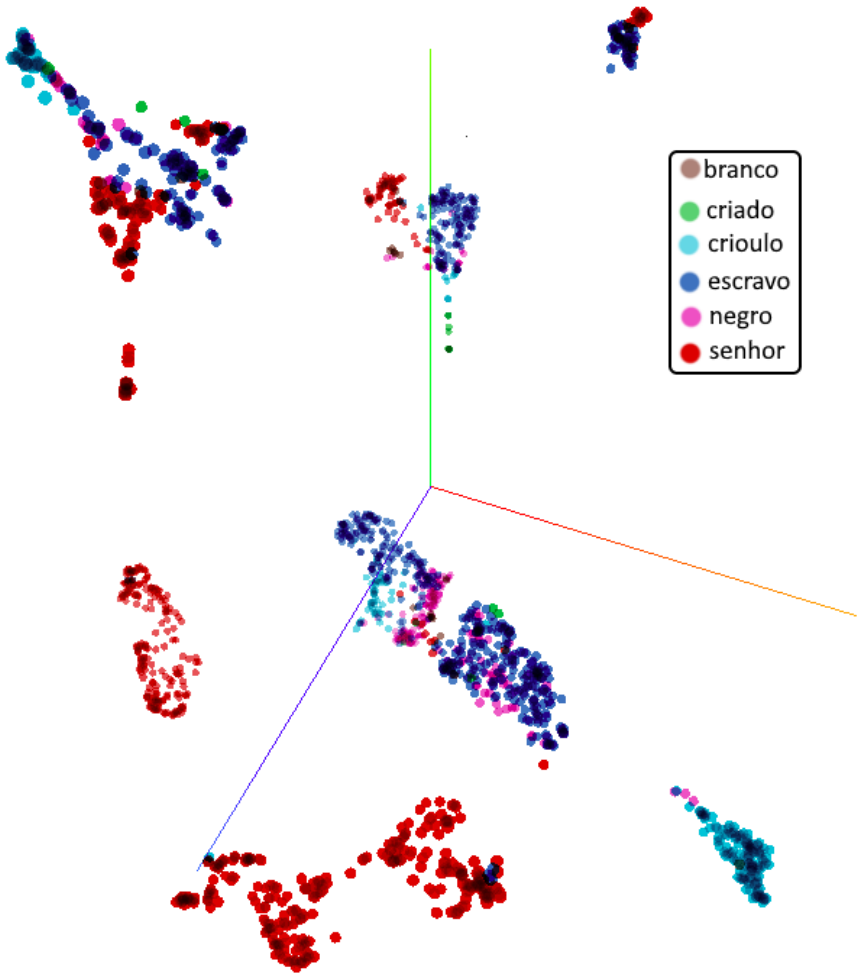


Fonte: o autor.

A Figura 17 apresenta a visualização em três dimensões gerada por projeção UMAP de *embeddings* de janelas de palavras de tamanho 3 de sentenças relevantes da obra *As Vítimas-*

Algozes. Nessa figura, podemos identificar 7 grupos bem definidos de pontos. Destacam-se à esquerda dois grupos na cor vermelha referente a janelas com a palavra central "senhor". Outros grupos com sobreposições são identificados nas cores azul-claro para o grupo "crioulo", marrom o grupo "branco", verde o grupo "criado", azul o grupo "escravo"e na cor rosa o grupo da palavra-alvo "negro".

Figura 17 – Projeção UMAP de embeddings de janelas com 3 palavras nas sentenças da obra *As Vítimas-Algozes*.

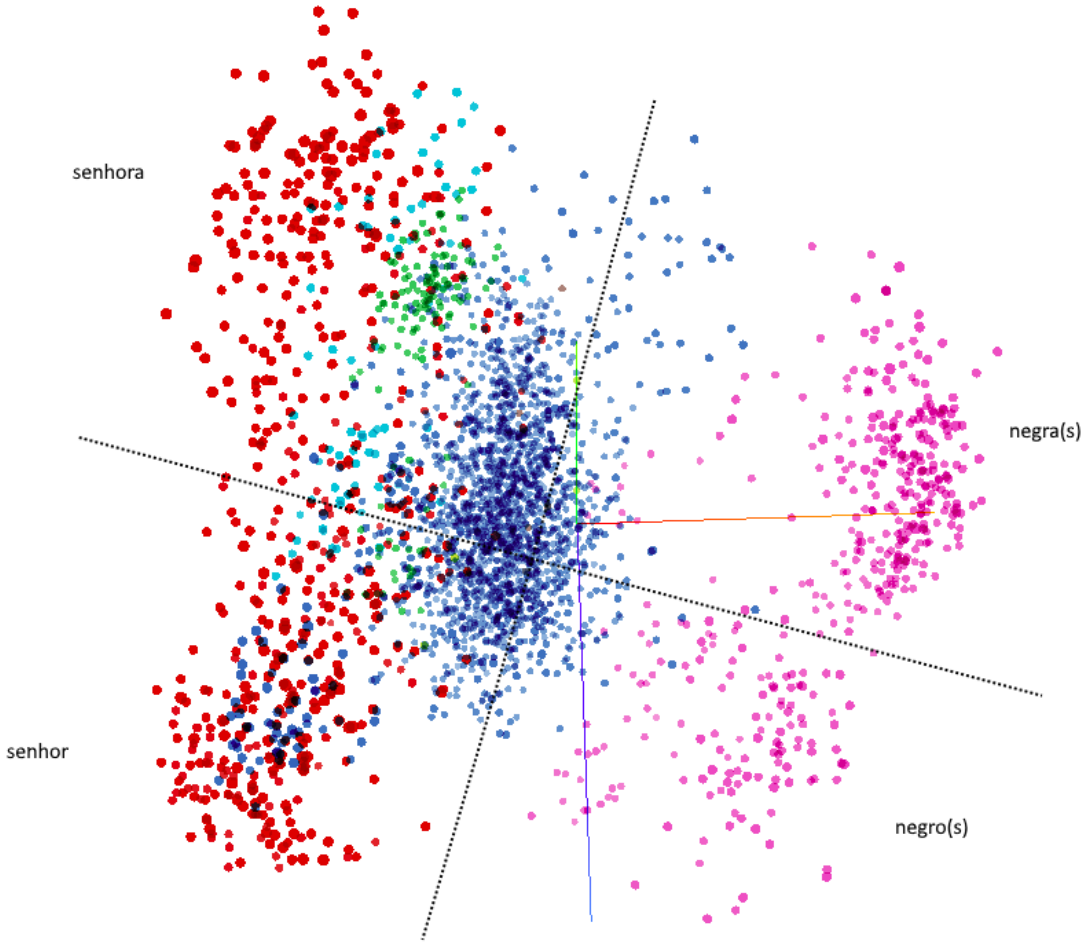


Fonte: o autor.

Na visualização dos *embeddings* de palavras das sentenças relevantes, optamos pelo recorte, pois a obra *As Vítimas-Algozes* apresenta um grande número palavras, não sendo possível identificar agrupamentos das palavras-alvo com as demais palavras da obra. Optou-se então por isolar e seleciona os *embeddings* segundo quadrantes para facilitar a visualização de um conjunto menor de pontos. A Figura 18 apresenta a visualização em três dimensões utilizando a técnica PCA do recorte do segundo quadrante dos *embeddings* de palavras das sentenças relevantes da obra *As Vítimas-Algozes*. Com isso identificou-se quatro grupos bem definidos, sendo eles: senhor (cor vermelha), negro (cor rosa), crioulo (verde-claro) e demais palavras (cor azul).

Outro ponto importante é a polaridade que foi apresentada, em que temos: polaridade à direita com o grupo "senhor" e à esquerda o grupo "negro". Além disso, ainda se tem a polaridade de gênero, onde acima temos as palavras do gênero feminino e abaixo as palavras do gênero masculino, destacando esse comportamento para os grupo "negro" e "senhor".

Figura 18 – Recorte da projeção PCA dos *embeddings* de palavras da obra *As Vítimas-Algozes*.



Fonte: o autor.

5.5 MINERAÇÃO DE PADRÕES MORFO-SEMÂNTICOS NOS TEXTOS

A mineração dos padrões morfo-semânticos nos textos literários usou a similaridade do cosseno entre *embeddings* de janelas de texto centradas em palavras-alvos com limiares (*thresholds*) de $0,8$ e $0,9$ para caracterizar similaridade , conforme utilizado por Sorato, Goularte e Fileto (2020). A Tabela 7 apresenta estatísticas das medidas de similaridade e distância entre *embeddings* de sentenças. Nessa Tabela, podemos verificar que 75% (terceiro quartil - Q3) das sentenças apresentam similaridade do cosseno superior a 0.8893. Este valor está dentro dos limites utilizados por Sorato, Goularte e Fileto (2020).

Tabela 7 – Estatísticas das medidas de similaridade e distância entre sentenças.

Estatísticas	cosseno	Euclidiana	Manhattan
#Sentenças	3.396.616	3.396.616	3.396.616
Média	0,8470	8,5226	212,9505
Desvio padrão	0,0560	1,8904	46,6709
Mínimo	0,4863	0,0000	0,0000
25% (Q1)	0,8129	7,0882	177,5473
50% (mediana)	0,8543	8,3964	210,0354
75% (Q3)	0,8893	9,7990	244,6087
Máximo	1,0000	18,7797	466,0917

Fonte: o autor.

A Tabela 8 apresenta as quantidades de sentenças, as médias de similaridade e distâncias, assim como os tamanhos das sentenças (em termos de número de tokens) contendo cada grupo de palavras-alvo. Note que tais sentenças têm tamanho médio entre 27 e 40 tokens e similaridade média entre pares de sentenças contendo cada grupo de palavras-alvo sempre maior que de $0,86$.

Tabela 8 – Quantidades e medias das distâncias, similaridade e tamanho dos grupos de sentenças com *threshold* maior ou igual 0,8.

Grupo	Qtd.	Média cos	Média Euc	Média Man	Tamanho médio
negro	298.497	0,8659	7,9626	199,4116	28
crioulo	360.003	0,8671	7,9298	198,5817	27
criado	27.105	0,8678	7,8531	197,1207	40
escravo	1.017.569	0,8736	7,6566	191,6855	37
senhor	1.012.351	0,8635	8,0975	202,5658	29
branco	32.951	0,8658	7,9589	198,5012	39

Fonte: o autor.

Abaixo listamos as 5 sentenças com maior número de ocorrências de palavras-alvo do grupo "negro". A primeira tem 3.204 sentenças cuja similaridade com ela é maior ou igual $0,8$.

1. "Chegados à sala de jantar o crioulo mostrou ao lado direito a porta do aposento de Hermano e de Florinda ; dois dos perversos , o Barbudo e um dos negros colocaram-se aos lados dessa porta : o outro negro recebeu a lanterna e seguiu a Simeão que avançou para a frente e entrou no quarto de dormir de sua senhora ."
2. "mas , pois que a do Pai-Raiol e de Esméria lhe aproveitava , reputou afortunada a compra que mantinha a consoladora sociedade do negro e da negra que se diziam amar ."
3. "– Sou negra escrava lançada no campo : animal solto e livre , se eu me desforrasse do desprezo em que meu senhor me abandona , abrindo a porta da minha senzala aos negros meus parceiros e do meu gosto , faria muito bem ."
4. "Simeão , esquecendo o golpe que recebera , e o sangue que do ombro lhe corria , deixou um dos negros na sala onde estavam os dois cadáveres e com o Barbudo e o outro negro

que levava a lanterna , voltou ao quarto da senhora assassinada , arrombou facilmente a gaveta da velha mesa , e apoderando-se de uma grossa chave , foi ao fundo do quarto , arrancou precipitado uma cortina de chita que cobria pequena parte da parede e mostrando em grande vão que havia nesta uma caixa de jacarandá chapeada de ferro , abriu rápido duas fechaduras , e escancarou a caixa que estava cheia de pequenos sacos contendo moedas de ouro e prata."

5. "É o santo negro que ajuda os diabos negros !"

A Tabela 9 apresenta estatísticas das comparações entre os *embeddings* de janelas abrangendo 3 tokens (palavra-alvo, um token à sua direita e outro à sua esquerda) dentro das sentenças. Note que 25% das sentenças apresentam similaridade do cosseno maior que 0.7403, portanto dentro dos limites utilizados por Sorato, Goularte e Fileto (2020). Assim, os padrões minerados com este patamar de similaridade ocorrem em 25% do conjunto de dados.

Tabela 9 – Estatísticos das medidas de similaridade e distância entre janelas de tamanho 3.

Estatísticas	**cosseno**	**Euclidiana**	**Manhattan**
#Janelas	2.972.522	2.972.522	2.972.522
Média	0,6859	14,3459	354,1851
Desvio padrão	0,0782	1,7994	43,5384
Mínimo	0,2316	0,0000	0,0000
25% (Q1)	0,6342	13,1743	325,9953
50% (mediana)	0,6893	14,3891	355,1178
75% (Q3)	0,7403	15,5826	383,8223
Máximo	1,0000	22,5513	541,8450

Fonte: o autor.

A Tabela 10 apresenta as quantidades de janelas de tamanho 3 para cada grupo de palavras-alvo e as médias de similaridade e distâncias entre seus *embedings*. Note que essas janelas têm em média similaridade entre seus *embedings* superior a $0,91$.

Tabela 10 – Quantidades e medias das distância, similaridade dos grupos de janelas de tamanho 3 com *threshold* maior ou igual $0,9$.

Grupo	**Qtd.**	**Média cos**	**Média Euc**	**Média Man**
negro	561	0,9129	7,7572	193,9294
crioulo	840	0,9131	7,5950	191,0693
escravo	780	0,9110	7,6593	191,3575
senhor	1.939	0,9152	7,6797	192,5278

Fonte: o autor.

A Tabela 11, lista as 5 janelas de sentenças com maior número de ocorrências de palavras-alvo do grupo "negro". Estas são janelas das sentenças relevantes (com palavras-alvo) com maior número de outras janelas similares também relevantes. A primeira janela da lista tem 23 janelas similares a ela. A repetição de conteúdos de janelas na terceira e na quarta linhas

lista ocorreu porque as janelas ali apresentadas pertencem a sentenças diferentes na obra que possuem *embeddings* contextualizados distinto. As quantidades de janelas similares a elas (com similaridade maior do que $0,8$) são 22 e 21, respectivamente.

Tabela 11 – Lista das 5 janelas de sentenças com maior quantidade de ocorrência similaridade entre os *embeddings*

Janela	**Fragmento da sentença**	**Qtd.**
"O negro conservava-se"	[...]" **O negro conservava-se** imóvel e como insensível."[...]	23
"O negro arrancou-se"	[...]" **O negro arrancou-se** dos braços da crioula, e fitando-a de novo, com olhar imponente de sua vontade, absoluto, imperioso disse ainda, dando à voz tom ameaçador: –"[...]	22
"o negro riu-se"	[...]"Daí a pouco **o negro riu-se**, olhando para o ventre da crioula."[...]	22
"O negro sacudiu"	[...]" **O negro sacudiu** com a cabeça, e tornou com voz comprimida e alterada:"[...]	21
"O negro riu-se"	[...]" **O negro riu-se** outra vez e disse:"[...]	21

Fonte: o autor.

A Tabela 12, apresenta as 23 janelas de sentenças com *embeddings* mais similares a janela "O negro conservava-se", nela é possível entender melhor o padrão minerado.

5.6 DISCUSSÃO

O conjunto de dados utilizado apresenta um vocabulário rico devido ao tipo de texto analisado, onde o escritor tende a não repetir a mesma frase ou palavra, aproveitando-se dos sinônimos e alternância para transmitir a informação desejada. A segmentação de sentenças (descrita na Seção 4.1) gerou sentenças grandes, com algumas contendo mais do que 250 tokens, além de sentenças com mais de uma palavra-alvo. Isso pode ter influenciado na mineração de padrões (tarefa descrita na Seção 4.6), que resultou em baixo número de padrões morfo-semânticos em volta das palavras-alvo encontradas.

O idioma português apresenta um vocabulário muito rico, podendo ter flexões para indicar conjugação verbal, grau do substantivo e outras. Isso possibilita ao autor descrever uma cena de várias formas. Além disso, há a possibilidade de alguma sentença não relevante (i.e., que não contenha palavra-alvo) apresentar um alto grau de similaridade com sentenças relevantes, constituindo um padrão a ser analisado.

Na tarefa de geração de *embeddings* (descrita na Seção 4.3) houve perda de informação devido ao grande volume de tokens desconhecidos. Talvez se utilizarmos um modelo do BERT treinado com os textos literários do mesmo período das obras analisadas, esta perda de informação seja minimizada, o que pode ter efeito positivo nas comparações do *embeddings* (descrita na Seção 4.5).

A tarefa de visualizar e selecionar *embeddings* (descrita na Seção 4.4) possibilitou identificar alguns bons agrupamentos. Principalmente as visualizações de *embeddings* de janelas em volta de algumas palavras-alvo relevantes mostraram agrupamentos interessantes. Por exemplo, como os apresentado na Figura 17, o agrupamento das janelas em torno de palavras-alvo do grupo "senhor"e do grupo "crioulo"são bem definidos. Todavia, o *Embedding Projetor* não permite replicar as visualizações geradas, devido às características do algoritmos que utiliza. O *Embedding Projetor* fornece um meio de salvar uma visualização gerando um *bookmark*. Entretanto, esse *bookmark* é estático e não possibilita quase nenhuma interação.

Finalmente, a configuração de hardware fornecida pelo *Google colaboratory* limitou os experimentos nas tarefas de gerar *embeddings* e minerar padrões, pois foram utilizados somente *embeddings* da última camada do BERT, para que as estruturas criadas nos experimentos não consumissem todos os 12 GB de memória disponibilizados. A ideia inicial era seguir a recomendação do Devlin et al. (2019) que sugere utilizar a concatenação das 4 últimas camadas do BERT que geraria *embeddings* com $4,096$ valores (dimensões). Isso pode ter comprometido as medidas de similaridade e distância e a mineração de padrões com tais medidas. Outro problema foi tempo de execução de cada experimento. Alguns experimentos demoraram cerca de 30 minutos até 2 horas e 40 minutos para serem executados. Mas o *Google colaboratory* encerra a sessão caso identifique inatividade do notebook por tanto tempo. Por conta destas limitações, os experimentos foram reduzidos a utilizar somente sentenças que apresentassem pelo menos uma palavra-alvo.

Tabela 12 – Lista das 23 janelas com *embeddings*
similares à janela "O negro conservava-se”

Janela	Fragmento de Sentença
"O negro ficou"	[...]"O negro ficou impávido; mas franziu as sobrancelhas."[...]
"O negro olhou"	[...]"O negro olhou suspeitoso, mas soberbo para a crioula, e viu a lascívia abrasando-lhe o rosto."[...]
"O negro apertou-lhe"	[...]"O negro apertou-lhe a mão e sentou-se à porta da senzala: a crioula imitou-o sentando-se a seu lado."[...]
"o negro ,"	[...]"como acabava de dizer o negro, muito mais bonita e elegante do que sua senhora."[...]
"O negro insistia"	[...]"O negro insistia ainda no conselho ou ordem que dera a Esméria, a qual continuava a fingir-se hesitante."[...]
"o negro fixou"	[...]"mas o negro fixou os olhos na ninhada de pintainhos, como se os quisesse absorver nas órbitas."[...]
"o negro ,"	[...]"– Dantes era melhor – disse o negro, sossegado."[...]
"O escravo preferia"	[...]"O escravo preferia ver talhar-se uma mortalha."[...]
"o negro que"	[...]"Esméria considerava, contemplava ansiosa o negro que, imóvel e de olhos fitos, mirava a ninhada infeliz."[...]
"O escravo não"	[...]"O escravo não precisava ser seduzido para encarregar-se da comissão: não tinha em estima o recato de sua senhora-moça; não quis, porém, receber a carta antes de entender-se com Lucinda."[...]
"o negro ;"	[...]"Teresa lembrou-se da impressão repulsiva que experimentara vendo o negro;"[...]
"O negro arrancou-se"	[...]"O negro arrancou-se dos braços da crioula, e fitando-a de novo, com olhar imponente de sua vontade,"[...]
"O negro deixava"	[...]"O negro deixava indiferentemente à mercê de Esméria a vida do senhor, a quem não segurava, nem empurrava."[...]
"O negro riu-se"	[...]"O negro riu-se outra vez e disse:"[...]
"O negro sacudiu"	[...]"O negro sacudiu com a cabeça, e tornou com voz comprimida e alterada:"[...]
"O negro afagou"	[...]"O negro afagou a crioula como costumava, insinuando-se possuído de paixão cada vez mais violenta."[...]
"o negro riu-se"	[...]"Daí a pouco o negro riu-se, olhando para o ventre da crioula."[...]
"O negro ria-se"	[...]"O negro ria-se de modo a causar pavor;"[...]
"o negro tornou-se"	[...]"Entretanto e por isso mesmo que o segredo desaparecera, **o negro tornou-se** mais exigente e aos domingos e dias santificados reclamava com renitência a companhia de Esméria,"[...]
"O negro encolheu"	[...]"O negro encolheu os ombros."[...]
"O negro riu-se"	[...]"O negro riu-se;"[...]
"O negro pôs-se"	[...]"O negro pôs-se a rir com o seu medonho riso: ele sabia que a crioula não era menos devassa que dantes."[...]
"O negro pareceu"	[...]"O negro pareceu indignado."[...]

Fonte: o autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, desenvolvemos uma abordagem baseada em PLN para minerar padrões morfo-semânticos em textos literários visando suportar classificação não supervisionada e análise semântica de discursos em torno de tópicos fornecidos, em um estudo de caso na área de literatura. Propusermos um algorítimo que utiliza o modelo chamado *BERTimbau* uma versão em português do BERT para o processamento de textos literários. O processo se inicia pela segmentação do texto em sentenças, POS-Tagging e lematização, selecionar sentenças relevantes, gerar instâncias candidatas, gerar visualização de *embeddings* de instâncias e, por fim, analisar a distribuição dos *embeddings* das instâncias.

Os resultados experimentais revelaram padrões determinados por similaridades entre sentenças. Sentenças contendo palavras-alvo de um mesmo grupo tendem a ficar próximas. Por exemplo, as sentenças que apresentavam à palavra-alvo "senhor"e "senhora" ficaram próximas quando utilizamos o *Embedding Projector*, e o mesmo fenômeno se repetiu com as visualizações das janelas. Porém, nas visualizações dos *embeddings* das palavras foi possível identificar que o conjunto de dados apresentou polaridade de gênero e de classe social.

Trabalhos futuros incluem: (*i*) aprimorar os algoritmos para mineração de padrões morfo-semânticos baseados em proximidade de *embeddings*, inclusive ao nível de palavras individualmente; (*ii*) treinar modelo de linguagem com textos de obras literárias do mesmo período das obras mineradas; (*iii*) explorar a técnica de janelas deslizantes para tentar gerar melhores *embeddings*; (*iv*) utilizar modelos de *word embeddings* mais recentes, tais como o *BLOOM*[1] e (v) investigar as possibilidades de aplicação da abordagem proposta a textos de outros domínios, tais como textos científicos, jurídicos e de literaturas atuais.

-

[1] https://huggingface.co/bigscience/bloom

REFERÊNCIAS

ARAúJO-JUNIOR, J. R. d. S. Mineração de poemas através de técnicas de processamento de linguagem natural. **-**, Universidade Federal de Campina Grande, 2021.

BASILE, V. et al. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In: **Proceedings of the 13th International Workshop on Semantic Evaluation**. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019. p. 54–63. Disponível em: https://aclanthology.org/S19-2007.

BRAZ-JUNIOR, O. d. O.; FILETO, R. Investigando coerência em postagens de um fórum de dúvidas em ambiente virtual de aprendizagem com o bert. In: **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. Porto Alegre, RS, Brasil: SBC, 2021. p. 749–759. ISSN 0000-0000. Disponível em: https://sol.sbc.org.br/index.php/sbie/article/view/18103.

CHEN, L.; VAROQUAUX, G.; SUCHANEK, F. A lightweight neural model for biomedical entity linking. In: **The Thirty-Fifth AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2021. (-, 35), p. 14.

CHOWDHARY, K. Natural language processing. **Fundamentals of artificial intelligence**, Springer, p. 603–649, 2020.

CORDEIRO, B. C. **BERT E WORD2VEC: UMA ANALISE INFERENCIAL E COMPUTACIONAL NA CLASSIFICACAO DE TEXTOS COM REDES NEURAIS CONVOLUCIONAIS**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2019.

DAVIDSON, T. et al. **Automated Hate Speech Detection and the Problem of Offensive Language**. arXiv, 2017. Disponível em: https://arxiv.org/abs/1703.04009.

DEVLIN, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: https://aclanthology.org/N19-1423.

GOULARTE, F. B. et al. MSC+: language pattern learning for word sense induction and disambiguation. **Knowl. Based Syst.**, v. 188, 2020. Disponível em: https://doi.org/10.1016/j.knosys.2019.105017.

JOLLIFFE, I. **Principal Component Analysis**. [S.l.]: Springer Verlag, 1986.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing (2Nd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009. ISBN 0131873210.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. Citeseer, 2007.

LEITE, J. A. et al. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. **arXiv preprint arXiv:2010.04543**, 2020.

MAATEN, L. van der; HINTON, G. Visualizing data using t-SNE. **Journal of Machine Learning Research**, v. 9, p. 2579–2605, 2008. Disponível em: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

MCINNES, L.; HEALY, J.; MELVILLE, J. **UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction**. arXiv, 2018. Disponível em: https://arxiv.org/abs/1802.03426.

MIKOLOV, T. et al. **Efficient Estimation of Word Representations in Vector Space**. arXiv, 2013. Disponível em: https://arxiv.org/abs/1301.3781.

MULLER, P.; BRAUD, C.; MOREY, M. ToNy: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In: **Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019**. Minneapolis, MN: Association for Computational Linguistics, 2019. p. 115–124.

OSTENDORFF, M. et al. Pairwise multi-class document classification for semantic relations between wikipedia articles. **arXiv preprint arXiv:2003.09881**, 2020.

PORTO, W. **Venda de livros digitais cresce 115% em três anos, mostra pesquisa**. 2020. Disponível em: https://www1.folha.uol.com.br/ilustrada/2020/08/venda-de-livros-digitais-cresce-115-em-tres-anos-mostra-pesquisa.shtml.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 3982–3992.

SHENG, D.; YUAN, J. An efficient long chinese text sentiment analysis method using bert-based models with bigru. In: IEEE. **2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)**. [S.l.], 2021. p. 192–197.

SHORTEN, C.; KHOSHGOFTAAR, T. M.; FURHT, B. Deep learning applications for covid-19. **Journal of Big Data**, Springer, v. 8, n. 1, p. 1–54, 2021.

SMILKOV, D. et al. Embedding projector: Interactive visualization and interpretation of embeddings. **arXiv preprint arXiv:1611.05469**, 2016.

SORATO, D.; GOULARTE, F. B.; FILETO, R. Short semantic patterns: A linguistic pattern mining approach for content analysis applied to hate speech. **International Journal on Artificial Intelligence Tools**, World Scientific, v. 29, n. 02, p. 2040002, 2020.

SORATO, D. et al. Análise de métodos e ferramentas para reconhecimento de palavras relevantes em microblogs. In: SBC. **Anais do XII Simpósio Brasileiro de Sistemas de Informação**. [S.l.], 2016. p. 345–352.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: pretrained bert models for brazilian portuguese. In: SPRINGER. **Brazilian conference on intelligent systems**. [S.l.], 2020. p. 403–417.

TENNEY, I.; DAS, D.; PAVLICK, E. BERT rediscovers the classical NLP pipeline. **CoRR**, abs/1905.05950, 2019. Disponível em: http://arxiv.org/abs/1905.05950.

VALLERIAN, A. **What Is Metric?**: Understanding metric for data scientist. 2021. Disponível em: https://towardsdatascience.com/what-is-metric-74b0bf6e862. Acesso em: 09 jul. 2022.

WASEEM, Z.; HOVY, D. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In: **Proceedings of the NAACL Student Research Workshop**. San Diego, California: Association for Computational Linguistics, 2016. p. 88–93. Disponível em: https://aclanthology.org/N16-2013.

WHAT is Euclidean And Manhattan distances in KNN? 2020. Disponível em: https://community.insaid.co/hc/en-us/articles/360052305633-What-is-Euclidean-And-Manhattan-distances-in-KNN-.

WILLRICH, R. et al. Capture and visualisation of text understanding through semantic annotations and semantic networks for teaching and learning. **J. Inf. Sci.**, v. 46, n. 4, 2020. Disponível em: https://doi.org/10.1177/0165551519849514.

WOLF, T. et al. Transformers: State-of-the-art natural language processing. In: **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**. [S.l.: s.n.], 2020. p. 38–45.

ZHANG, Z.; ZHAO, H.; WANG, R. Machine reading comprehension: The role of contextualized language models and beyond. **arXiv preprint arXiv:2005.06249**, 2020.

APÊNDICE A – ARTIGO DO TABALHO

Mineração de padrões morfo-semânticos em textos literários com o BERT

Leandro da Silveira Dias[1], Osmar de Oliveira Braz Junior[1], Renato Fileto[1]

[1]Universidade Federal de Santa Catarina – PPGCC
Florianopolis, Santa Catarina, Brazil.

{leandro.dias}@grad.ufsc.br, {osmar.braz}@posgrad.ufsc.br

{renato.fileto}@inf.ufsc.br

Abstract. *The automated semantic analysis of discourses around the presentation of interest in textual documents is an open problem, with several practical applications. This work proposes to develop new techniques and algorithms to mine morpho-semantic patterns of discourses centered on controls. The proposal is being developed and evaluated in a case study in the field of Brazilian literature. The results will be quantitatively evaluated, in terms of the distribution of instances of mined patterns in document collections and, as far as possible, comparison with human performance in identifying patterns and classifying texts*

Keywords: *Morpho-semantic Patterns in Texts. Natural Language Processing (NLP). Text Mining. Embeddings. Literary texts.*

Resumo. *A análise semântica automatizada de discursos em torno de tópicos de interesse em documentos textuais é um problema ainda em aberto, com diversas aplicações práticas. Este trabalho se propõe a desenvolver novas técnicas e algoritmos para minerar padrões morfo-semânticos de discursos centrados em tópicos. A proposta está sendo desenvolvida e avaliada em um estudo de caso na área de literatura brasileira. Os resultados serão avaliados quantitativamente, em termos da distribuição das instâncias dos padrões minerados nas coleções de documentos e, na medida das possibilidades, comparação com o desempenho humano na identificação dos padrões e classificação dos textos.*

Palavras-alvo: *Padrões Morfo-semânticos em Textos. Processamento de Linguagem Natural (PLN). Mineração de Textos. Embeddings. Textos literários.*

1. Introdução

O consumo de informação na forma de texto em meio digital vem aumentando rapidamente. Segundo o jornal Folha de São Paulo [Porto 2020], houve um aumento de 115% na venda de *E-Books* entre os anos de 2016 a 2019. Mais recentemente, a pandemia do COVID-19 potencializou o consumo desses textos digitais. O grande volume de textos torna difícil a análise manual dos seus conteúdos, o que tem levado ao desenvolvimento de soluções utilizando ferramentas para PLN, mineração de textos (do inglês, TM) e ciência de dados (do inglês, DS).

Técnicas e ferramentas de PLN têm sido utilizadas com sucesso em diversas tarefas, incluindo reconhecimento de entidades nomeadas (do inglês, *Named Entities Recognition* - NER) para encontrar menções a entidades relacionadas à COVID-19 e auxiliar na resposta à pandemia [Shorten et al. 2021], ligação de entidades (do ingles, *Entity Linking*) para vincular entidades na área de biomedicina [Chen et al. 2021] e classificação morfossintática de palavras (do inglês, *Part-Of-Speech* - POS-*Tagging*) que, juntamente com embeddings de palavras e aprendizado de máquina, permite detectar e analisar discursos de ódio em textos de mídias sociais [Sorato et al. 2020, Leite et al. 2020]. Atualmente, *embeddings* contextualizados têm possibilitado ganhos consideráveis de desempenho em tarefas como categorizar textos de portais de noticias [Sheng and Yuan 2021], analisar coerência em textos [Braz-Junior and Fileto 2021] e sumarizar textos longos [Sheng and Yuan 2021].

1.1. Descrição do Problema

A Biblioteca Digital de Literatura de Países Lusófonos (BLPL)[1], além de oferecer acesso a metadados e ao próprio texto de um grande número de obras literárias em língua portuguesa, é um exemplo de coleção de textos digitais sobre a qual são desenvolvidas, exercitadas e avaliadas estratégias para ensino de literatura, análise literária e leitura distante. Seu acoplamento ao Moodle e à ferramenta de anotação semântica DLNotes [Willrich et al. 2020], entre outras, permite realizar, por exemplo, tarefas de anotação semântica de textos. Este ambiente constitui um excelente campo experimental de novas tecnologias para mineração e análise de textos, incluindo análise semântica de discursos. O Núcleo de Pesquisa em Informática, Literatura e Linguística (NUPILL-UFSC)[2], que desenvolveu e mantém a BLPL e o DLNotes, entre outros artefatos, inclui especialistas em literatura, linguagem humana e computação, que unem esforços para inovar na confluência dessas áreas, além de colaboradores diversos para auxiliar na definição de soluções e avaliação do seu desempenho. Ao serem apresentados à possibilidades de efetuar análise semântica automatizada de discursos mediante mineração de padrões em torno de tópicos fornecidos, os especialistas em literatura e linguística do NUPILL sugeriram como primeira tarefa para estudo de caso em literatura minerar padrões semânticos de discursos em torno do tópico ***escravidão*** (que pode ser expresso por palavras como *escravidão, escravo, negro, mulato* e *senzala*) em obras selecionadas de alguns autores.

O problema tratado neste trabalho, em colaboração com pesquisadores do NUPILL, é minerar padrões semânticos de discurso em torno de tópicos fornecidos. Um padrão semântico refere-se a um conjunto de instâncias de componentes textuais (e.g. sentenças) com palavras de sentido similar em torno de palavras que definam o tópico focado. A título de exemplo, as seguintes sentenças, extraídas da obra *As Vítimas-Algozes*, de Joaquim Manuel de Macedo, publicada em 1869, constituem instâncias do mesmo padrão semântico em torno do tópico ***escravidão***, o qual também pode é denotado pela palavra ***escravo*** nessas sentenças:

1. *"Em falta de pundonor[1] e de vergonha[2], que a* ***escravidão*** *não comporta, o* ***escravo*** *tem o rancor[3] e o desejo da vingança[4]."*

[1] https://www.literaturabrasileira.ufsc.br/public/index.php
[2] https://nupill.ufsc.br/

2. *"Nas pontas do açoite[1] está o emblema do rancor[3] do* ***escravo****: às vezes há nas pontas do açoite[2] marcas de sangue[4]."*

O padrão semântico de discurso é determinado pela compatibilidade semântica entre palavras usadas em torno do tópico ***escravidão*** nas duas sentenças acima (instâncias do padrão). A palavra *pundonor*[1] da primeira sentença pode ser considerada semanticamente relacionada à *açoite*[1] na segunda sentença, assim como *vergonha*[2] a *açoite*[2] e *vingança*[4] a *sangue*[4]. Além disso, a palavra *rancor*[3] está em ambas as sentenças. Note que o discurso em torno do tópico pode incluir um número distinto de palavras (com significado relevante) em cada sentença, cada uma dessas palavras pode ser compatível com várias palavras da outra sentença e não precisa haver correspondência biunívoca entre palavras das duas sentenças para haver o padrão. Este padrão é definido pelo alinhamento semântico dos discursos das duas sentenças, que pode ser medido como similaridade entre palavras, possivelmente mas não necessariamente considerando também a compatibilidade de suas classes morfo-sintáticas e/ou morfologia, pois também podem influenciar no sentido das palavras, assim como o contexto onde elas são usadas. Assim, podemos denominar tais padrões como morfo-semânticos.

subsectionObjetivos

O objetivo geral deste trabalho é minerar padrões morfo-semânticos em textos literários, visando suportar classificação não supervisionada e análise semântica de discursos em torno de tópicos fornecidos, em um estudo de caso na área de literatura. Para alcançá-lo, é necessário atingir os objetivos específicos abaixo relacionados.

1. Estudar, selecionar e dominar técnicas e ferramentas do estado da arte Processamento de Linguagem Natural (PLN) necessárias para minerar padrões morfo-semânticos em textos.
2. Desenvolver e avaliar novos algoritmos eficientes e efetivos para minerar padrões morfo-semânticos em torno de tópicos fornecidas por usuários especialistas de domínio usando técnicas e ferramentas de PLN estudadas e selecionadas.
3. Analisar a distribuição das instâncias de padrões, classes morfossintáticas e sentidos das palavras envolvidas em textos literários, visando classificação e análise semântica de discursos de acordo com as ocorrências de tais padrões.

1.2. Metodologia

Inicialmente, serão realizados estudos sobre o estado da arte em tarefas e ferramentas de PLN que podem ser usadas na preparação dos dados textuais a serem minerados, nos próprios algoritmos de mineração de padrões morfo-semânticos e na avaliação dos seus resultados. Tais tarefas incluem normalização de texto (tokenização, *stemming*, etc.), classificação morfossintática (*PoS-Tagging*), reconhecimento de entidades nomeadas, ligação de entidades e cálculo de similaridade entre *embeddings* de palavras. Posteriormente, serão realizadas análises qualitativas e quantitativas das distribuições de palavras presentes nos textos a serem minerados, suas classes morfossintáticas, distâncias entre seus respectivos *embeddings* e outras medidas. Tais análises visam preparação adequada dos dados e definição de parâmetros para mineração de padrões morfo-sintáticos, incluindo, entre outros, seleção de tópicos e palavras que os representem, além de funções de similaridade ou distância semântica e limiares a serem considerados na determinação de palavras compatíveis.

Para resolver o problema de minerar automaticamente padrões morfo-semânticos, podem ser utilizadas diversos alternativas de medidas de similaridade e critérios adicionais para determinar compatibilidade de palavras. Neste trabalho exploramos *embeddings* contextualizados de palavras para determinar similaridade (e portanto compatibilidade) semântica de palavras, além de seus *POS-Tags* (e.g., substantivo, verbo), nos algoritmos que estamos desenvolvendo, adaptando e avaliando para minerar automaticamente os padrões morfo-semânticos em torno de tópicos fornecidos por usuários especialistas do domínio de literatura. Descrições desses artefatos são apresentadas em detalhes nas seções 2, 3 e 4.

Finalmente, será realizada a avaliação dos custos computacionais e da qualidade dos resultados obtidos com os algoritmos desenvolvidos. Será preparado um artigo com descrições dos algoritmos desenvolvidos com links para código-fonte no GitHub e demonstrações de seu funcionamento em notebooks do Google Colab, além das tarefas efetuadas nos experimentos, dados utilizados, configurações de parâmetros e análise dos resultados obtidos. A distribuição das instâncias dos padrões minerados nas coleções de documentos será avaliada quantitativamente, com o objetivo de caracterizar textos de diferentes autores, épocas e movimentos. Os custos computacionais serão avaliados em termos de tempo de processamento e uso de memória. Além disso, na medida das possibilidades, os resultados dos algoritmos serão confrontados com a apreciação humana na identificação dos padrões, classificação dos textos e análise semântica dos discursos.

1.3. Estrutura do trabalho

O restante deste trabalho está estruturado como se segue. A Seção 2 descreve os fundamentos utilizados no trabalho e necessários ao seu entendimento. A Seção 3 discute os trabalhos relacionados e compara as características do trabalho aqui proposto com o estado-da-arte. A Seção 4 descreve a proposta para minerar padrões morfo-semânticos em textos literários, cujo objetivo é propiciar meios para analisar e classificar tais de acordo com a semântica dos discursos expressa nesses padrões. A Seção 5 delineia o plano de experimentos para avaliar a proposta, reporta experimentos iniciais e discute seus resultados. Finalmente, a Seção 6 apresenta as conclusões parciais, o cronograma de atividades para finalizar o trabalho e enumera temas para trabalhos futuros.

2. Fundamentos

Esta seção primeiro fornece uma visão geral de Processamento de Linguagem Natural (PLN), incluindo conceitos fundamentais, aplicações e ferramentas. Em seguida, descreve as tarefas de PLN utilizadas no desenvolvimento da solução proposta neste trabalho. Posteriormente, discute *embeddings* de palavras, os principais modelos disponíveis para gerá-los, suas propriedades, técnicas de *pooling* para compor embeddings de componentes textuais maiores como sentenças, e cálculo de funções de distância e similaridade usando *embeddings*. O foco é no modelo contextualizado de linguagem BERT e a sua variação $BERTimbau$ pré-treinada para o português brasileiro atual. Finalmente, discute a principais funcionalidades da ferramenta *Embedding projector*, utilizada para visualizar *embeddings* manipulados neste trabalho.

2.1. Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN), segundo [Jurafsky and Martin 2009], é um ramo da inteligência artificial que visa dar ao computador a capacidade de processar lin-

guagem humana. Esse campo de estudo é interdisciplinar e por esse motivo ele pode ser denominado por outros nomes, tais como: processamento de fala e linguagem, tecnologia da linguagem, linguística computacional e reconhecimento e síntese de fala. O objetivo é fazer com que os computadores possam executar tarefas envolvendo linguagem humana, possibilitando a comunicação humano-máquina, melhorando a comunicação entre humanos ou simplesmente fazendo processamento útil de texto e/ou fala. [Chowdhary 2020] define PLN como um conjunto de métodos computacionais para realizar análise automática e representação das linguagens humanas. Algumas das aplicações mais comuns de PLN são: classificação de texto, tradução automática, extração de informações, aquisição de conhecimento, sumarização automática de textos e resposder perguntas estipuladas em linguagem natural (em inglês *Question Answering - QA*).

Atualmente há diversos kits de ferramentas para PLN, tais como: *spaCy*[3], *Stanza*[4] *FreeLing*[5] e *LX-Center*[6]. Tarefas de PLN suportadas por tais kits incluem tokenização, *Part-Of-Speech - POS-tagging*), *stemming* ou lemmatização, segmentação de sentenças (em inglês *Sentence Boundary Detection* - SBD), reconhecimento de entidades nomeadas (em inglês *Named Entity Recognition - NER), ligação de entidades (em Entity Linking - EL ou NED (Disambiguation)) e determinação de similaridades.* Este trabalho usa o *spaCy*, uma biblioteca de código aberto e gratuita com métodos e modelos pré-treinados em língua portuguesa para realizar tarefas como as citadas anteriormente. As subseções a seguir descrevem as de tarefas PLN usadas no desenvolvimento deste trabalho.

Normalização de texto

A normalização de texto é uma fase de pre-processamento de texto que engloba a realização de várias tarefas do PLN visando padronizar palavras do *corpus* a ser analisado. Essa padronização pode incluir tarefas como converter o texto para letra maiúscula ou minúscula e remoção de acentos, caracteres especiais e repetições. A tarefa de normalização de texto visa tornar a sua análise mais precisa, por remover possíveis ruídos no momento da análise estatística ou do modelo utilizado. Como exemplo, considere o texto a seguir extraído do livro *As Vítimas-Algozes*, de Joaquim Manuel de Macedo, seguido de sua normalização.

Trecho original: *"Nas pontas do açoite está o emblema do rancor do escravo: às vezes há nas pontas do açoite marcas de sangue"*

Trecho normalizado: *"nas pontas do acoite esta o emblema do rancor do escravo: as vezes ha nas pontas do acoite marcas de sangue"*

Note que no trecho normalizado, ocorreu a conversão de todas as letras maiúsculas para minuscula (e.g "Nas"para "nas"), bem como a remoção dos acentos (e.g "está"para "esta") e caracteres especiais (e.g "açoite"para "acoite"). Repetições (e.g "...."para ".") também são removidas quando ocorrem no texto original.

[3]https://spacy.io/
[4]https://stanfordnlp.github.io/stanza/
[5]https://nlp.lsi.upc.edu/freeling/
[6]http://lxcenter.di.fc.ul.pt/

Tokenização

A tokenização separa os *tokens* (palavras e sinais representativos) do texto. Métodos de tokenização aplicam regras ou modelos que variam com o idioma. O exemplo a seguir mostra o texto do exemplo anterior com cada token sublinhado sublinhados.

Trecho tokenizado ”nas pontas do acoite esta o emblema do rancor do escravo : as vezes ha nas pontas do acoite marcas de sangue”

Stemming ou lematização

As tarefas de *Stemming* e lematização servem para reduzir cada palavra à sua forma raiz, reduzindo o ruído devido a flexos (gênero, número, conjugação verbal, etc.). A lematização retorna o lema da palavra, (e.g. o lema de ”pontas”é ”ponta”), enquanto a tarefa de *Stemming* retorna o seu radical chamado em inglês *stem* (e.g. o radical de ”pontas”é ”pont”).

Detecção de Limites de Sentenças

A tarefa de Detecção de Limites de Sentenças (SBD), ou simplemente segmentação de sentenças, realiza o processo que quebra o texto em sequências de sentenças conforme a sinais de pontuação, como o ponto final. A título de exemplo, consideremos outro fragmento de texto do livro *As Vítimas-Algozes*, de Joaquim Manuel de Macedo a seguir:

Trecho original *”Excelente crioulo! Como ama a seu senhor! Há poucos assim. As aparências dissimulavam os sentimentos do escravo.”*

A tarefa de segmentação de sentenças divide este texto texto em 4 sentenças:

1. - Excelente crioulo!”
2. ”Como ama a seu senhor!”
3. ”Há poucos assim.”
4. ”As aparências dissimulavam os sentimentos do escravo.”

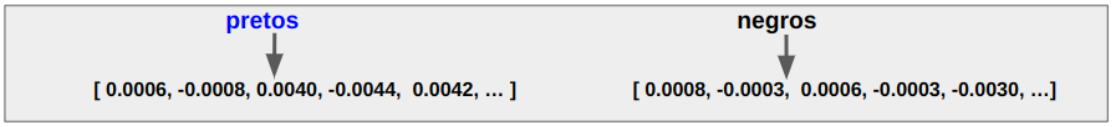
PoS-Tagging

A anotação morfossintática das palavras, usualmente denotada pelo termo em inglês *Part-Of-Speech Tagging* ou abreviadamente *POS-Tagging*, categoriza cada token de um corpus de acordo com sua classe morfossintática (verbo, adjetivo, etc.). Diferentes ferramentas podem utilizar diferentes conjuntos de classes morfossintáticas, os quais podem ser mais ou menos detalhados. Métodos e modelos para realizar *POS-Tagging* usualmente levam em consideração o contexto onde os tokens ocorrem no texto ou fala para tentar atribuir corretamente uma etiqueta/rótulo de classe morfossintática a cada token. Segundo [Sorato et al. 2016], é de grande importância que seja precisa a classificação dos elementos morfossintáticos de uma sentença. *POS-Tags* errôneos podem ocasionar erros de processamento subsequentes, porque diversas outras tarefas de PLN e mineração de texto dependem de *POS-Tagging* correto.

2.2. Embeddings

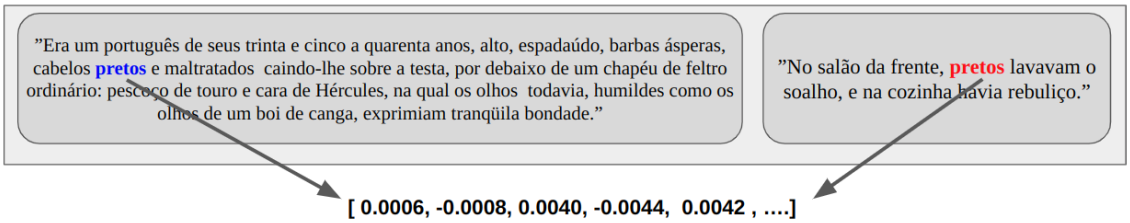
Um *embedding* de palavra pode ser definido a grosso modo como uma representação de uma palavra por meio de um vetor usualmente com centenas de dimensões. Cada dimensão de um tal vetor contém um número real, geralmente, entre -1 e 1 [Cordeiro 2019]. A representação vetorial é criada de tal modo que palavras com sentido similares ficam próximas umas das outras no espaço vetorial multidimensional, entre outras propriedades que podem ser capturadas. A Figura 1 ilustra *embeddings* de duas palavras distintas mas cuja semântica pode ser considerada similar, dependendo do contexto em que são usadas: "pretos"e "negros".

Figura 1. Palavras e seus *embeddings*



Os modelos de *embedding* mais tradicionais são *Word2Vec* [Mikolov et al. 2013] e GloVe [Zhang et al. 2020]. Muitas tarefas de PLN têm se beneficiado do seu uso. Porém, esses modelos tradicionais têm uma única representação vetorial para cada palavra, mesmo que seus significado ou mesmo classe morfossintática mude em diferentes contextos onde é usada, como ilustrado na Figura 2.

Figura 2. *Embeddings* de uma mesma palavra em frases diferentes



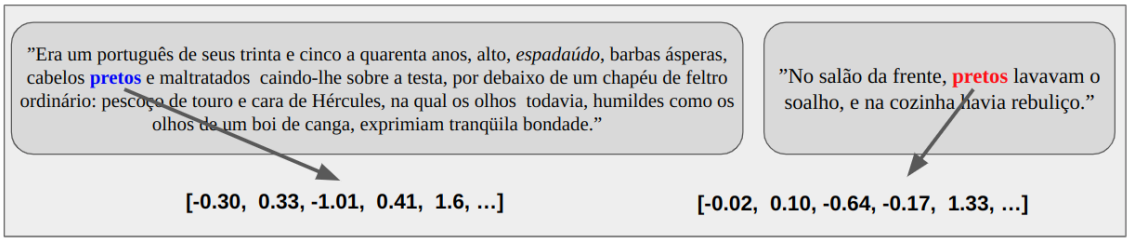
Na Figura 2 é possível ver que a palavra **"pretos"** tem sentidos distintos nas duas sentenças. Na sentença da esquerda, tal palavra faz o papel de adjetivo e tem o sentido de cor, enquanto na sentença da direta o sentido faz papel de substantivo com o sentido de indivíduo humano. Assim, as nuances de significado não são capturadas na representação vetorial. Modelos contextualizados de linguagem atuais permitem contornar este problema gerando diferentes *embeddings* para um mesmo token, pois consideram o contexto onde as palavras estão inseridas.

2.2.1. Modelos Contextualizados de Linguagem

Estudos recentes procuram meios de realizar a compreensão da comunicação em linguagem natural de forma automática através da aplicação de MCL (do inglês, *Contextualized Language Model*) [Zhang et al. 2020]. Os atuais MCLs costumam se baseados em rede neural bidirecional que possibilita capturar o contexto em que uma palavra ocorre em

um texto, e assim o seu significado específico naquele contexto. MCLs permitem ajustar *embeddings* de um mesmo léxico de acordo com o contexto, gerando representações diferentes para uma mesma palavra de modo a capturar nuances de significado em diferentes contextos de uso, como ilustrado pela Figura 3.

Figura 3. Palavras e seus *embeddings* contextualizados



BERT: *Bidirectional Encoder Representations from Transformers*

O BERT (acrônimo do inglês, *Bidirectional Encoder Representations from Transformers*) [Devlin et al. 2019] é um MCL constituído de uma rede neural profunda com processamento bidirecional. Ele é pré-treinado em um corpus de texto que ultrapassa 33 milhões de documentos textos não rotulados de cunho geral. Porém, permite ajustes finos com a adição de apenas uma camada de saída, visando otimizar o seu desempenho em tarefas de classificação de texto, reconhecimento de entidades nomeadas, em certos tipos de corpus como textos em blogs, diálogos entre tutores e estudantes [Devlin et al. 2019]. O BERT apresenta ótimos resultados na descoberta de estrutura sintática e semântica dentro de uma sentença [Tenney et al. 2019] em língua inglesa. Embeddings contextualizados como os gerados pelo BERT são úteis na solução de diversas tarefas de PLN, incluindo sumarização, classificação de textos e sistemas de diálogo. O BERT também tem propiciado ganhos consideráveis de desempenho quando ajusta pra efetuar tarefas de PLN, tais como segmentação [Muller et al. 2019] e comparação semântica de sentenças [Reimers and Gurevych 2019] e classificação de documentos [Ostendorff et al. 2020].

Trabalhos anteriores do nosso grupo de pesquisa usaram embeddings não contextualizados para a mineração de padrões morfo-semânticos em textos curtos, tais como postagens em mídias sociais [Sorato et al. 2016, Sorato et al. 2020], visando apoiar a análise de discursos. Textos longos, tais como os de literatura, têm mais informação de contexto. Assim, *embeddings* contextualizados têm potencial de contribuir para a qualidade da mineração de padrões morfo-semânticos nesses textos. Desta forma, este trabalho investiga o uso do BERTimbau [Souza et al. 2020], uma versão do BERT pré-treinada para língua portuguesa na mineração padrões morfossintáticos nas obras literárias. Este trabalho gera *embeddings* contextualizados usando o $BERTimbau$ e os utiliza para calcular distâncias ou similaridades em algoritmos de mineração de padrões morfo-semânticos em textos. O BERTimbau foi escolhido devido à disponibilidade gratuita de seus modelos pré-treinados, inclusive em língua portuguesa, e por conveniência do seu uso no *Google Colaboratory*, que disponibiliza acesso direto aos modelos através de bibliotecas específicas. O BERTimbau, assim como o BERT original, está disponível

em dois tamanhos: $BERTimbau_{base}$ com 12 níveis e 110 milhões de parâmetros e $BERTimbau_{large}$ com 24 níveis e 335 milhões de parâmetros.

2.3. Embedding projector

O *Embedding Projector* [Smilkov et al. 2016] é uma ferramenta de visualização de *embeddings* em duas ou três dimensões que ajuda a interpretar modelos de aprendizado de máquina que dependem de *embeddings*. A ferramenta permite explorar a vizinhança de pontos representando *embeddings* individuais, analisar a distribuição global dos pontos e investigar vetores semanticamente significativos no espaço. Possibilita realizar análises visuais das distribuições dos *embeddings* e buscas em formato de texto para testar hipóteses. O *Embedding Projector* é implementado como uma aplicação Web sobre a plataforma do *TensorFlow* para visualizar qualquer conjunto de *embeddings*, de qualquer dimensionalidade, fornecido através da plataforma ou em formato texto. O *Embedding Projector* oferece quatro métodos para reduzir a dimensionalidade de um conjunto de dados: Análise do componentes principais (PCA), Incorporação de vizinhos estocásticos distribuídos t (T-SNE), Aproximação e Projeção de Manifold Uniforme para Redução de Dimensões (UMAP) e Projeção customizável (CUSTOM).

3. Trabalhos relacionados

A Tabela 1 fornece um resumo comparativo das propostas selecionadas e analisadas. Os trabalhos são ordenados cronologicamente nas linhas da tabela. Eles são comparados conforme os aspectos das soluções de análise de textos que consideramos mais relevantes, que aparecem nas colunas. A primeira coluna (**Autor e Ano**) define o nome dos autores do trabalho avaliado e o ano de publicação. A segunda coluna (**Área de Aplicação**) refere-se à origem dos textos avaliados. A terceira coluna (**Objetivos**) refere-se ao que se pretende alcança com os estudos propostos. A quarta coluna re refere ao tipo de ***embedding*** utilizado por cada trabalho. Na quinta coluna (**Abordagem**) indica as técnicas utilizadas. Por fim, a coluna **Ferramenta** referencia as bibliotecas/ ou *frameworks* utilizados.

Grande parte dos trabalhos analisados não usam *embeddings* contextualizados. [Sorato et al. 2020] utiliza *embeddings* estáticos como GloVe para minerar padrões em textos, enquanto [Araújo-Junior 2021] e [Goularte et al. 2020] não usam *embedding* nenhum e o [Braz-Junior and Fileto 2021] utiliza *embeddings* contextualizados para classificador binário e um mensurador de (in)coerência em documentos. Nossa proposta pretende explorar representações de linguagem com base nos *embeddings* contextualizados produzidos pelo BERT, pois possuem representações que consideram o contexto em todas as direções. Além disso, o BERT permite ajuste-fino para tarefas *downstream*. Outra característica dos trabalhos relacionados é o fato deles manipularem textos curtos quando comparados aos textos literários considerados no nos presente trabalho, que pelo seu tamanho maior têm muito mais informação de contexto. Isso sugere que o uso do BERT, que é capaz de explorar esses contextos, pode contribuir na tarefa de encontrar padrões recorrentes envolvendo uma palavra-alvo em tais textos.

4. Processo geral e algoritmo para minerar padrões em textos

Esta seção descreve as etapas do processo proposto neste trabalho para minerar padrões morfo-semânticos em textos. A Figura 4 ilustra o fluxo de informação proposto para

Tabela 1. Quadro comparativo dos trabalhos relacionados

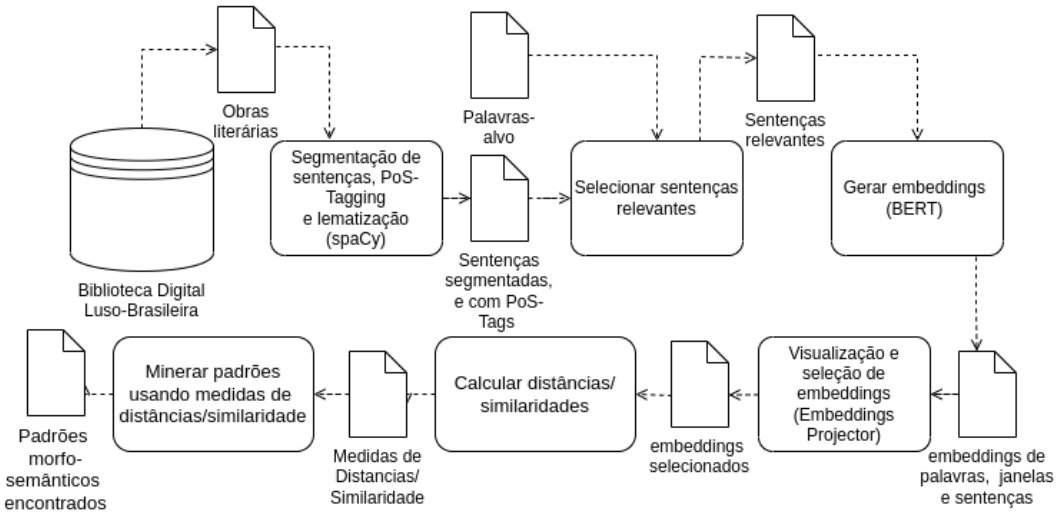
Trabalho	Textos processados	Objetivos	Casamento semântico	Tarefas PLN	Ferramenta de PLN
[Sorato et al 2020]	tweets	Minerar padrões para analisar e classificar discursos	GloVe	POS-*Tagging*	spaCy, *scikit-learn*
[Goularte et al 2020]	postagens em mídias sociais	Minerar padrões para desambiguar palavras	classes de entidades reconhecidas	POS-*Tagging*, NER	*FreeLing*, *WordNet*
[Araújo-Junior 2021]	Textos literários (Poemas)	Minerar Padrões	N/A	*POS-Tagging* correlação *tau* de *Kendall*, GSDMM, NPMI	*nlpnet*, *scipy*
[Braz-Junior and Fileto et al 2021]	perguntas de QA	Analisar coerência de discursos	BERT	*POS-Tagging*	*nlpnet*, *scipy*, Google Colab
Nossa proposta	**Textos Literários**	**Minerar padrões para analisar e classificar discursos**	**BERTimbau**	*POS-Tagging*	**spaCy, Google Colab**

minerar os padrões em obras literárias da Biblioteca Digital Luso-Brasileira[7], embora o processo proposto possa ser aplicado a textos de diversas outras fontes. O processo se inicia com o uso do SpaCy para efetuar segmentação do texto em sentenças, POS-*Tagging* e lematização. Posteriormente, seleciona sentenças relevantes, isto é, que contenham palavras alvo, em torno das quais se deseja minerar padrões e analisar discursos. O $BERTimbau$ é usado para gerar embeddings das palavras contidas nas sentenças selecionadas e o *Embedding Projector* para visualizar a distribuição desses *embeddings* e selecionar aqueles adequados para a mineração de padrões. Finalmente, os *embeddings* selecionados são comparados usando o algoritmo de [Sorato et al. 2020] e outros mais adequados a textos literários desenvolvidos no âmbito deste trabalho. Tais algoritmos avaliam a similaridade semântica das palavras usadas nos discursos em torno das palavras alvo para identificar os padrões semânticos, isto é instâncias de texto com os mesmo

[7]`https://www.literaturabrasileira.ufsc.br/`

sentidos em torno das mesmas palavras alvo, embora muitas vezes usando construções léxicas e sintáticas distintas.

Figura 4. Mineração de padrões morfo-semânticos em textos.



4.1. Segmentação do texto em sentenças, *POS-Tagging* e lematização

A primeira etapa do processo proposto é o pre-processamento dos textos. Nesta etapa primeiro se faz a padronização dos textos, mediante remoção de certos caracteres especiais (e.g.,$\backslash n$,$\backslash xad$) utilizados nas pagina da internet, remoção das repetições de espaços em branco, pontuações e símbolos (e.g., ponto final (.), ponto interrogação (?), ponto exclamação (!), traço ($-$)). Os textos padronizados são então submetidos à biblioteca *spaCy* para realizar as tarefas de tokenização, segmentação dos textos em sentenças, lematização e POS-*Tagging*. Após o término desses procedimentos, dados são armazenados utilizando uma estrutura semi-estruturadada, contendo id de cada sentença, lista de seus tokens, lista de lemmas, lista de POS-*Tags*, lista de verbos e a própria sentença. Esses dados são utilizadas para facilitar a seleção das sentenças relevantes para mineração de padrões.

4.2. Selecionar sentenças relevantes

A tarefa de selecionar sentenças relevantes é realizada utilizando os dados gerados na etapa anterior (descrita na Subseção 4.1). Uma sentença é selecionada se contém ao menos uma das palavras-alvo fornecidas pelos especialistas do domínio. O casamento desconsidera maiúsculas e minúscula (i.e., não é *case-sensitive*). As palavras-alvo estão separadas em grupos contendo as flexões consideradas de cada uma:

- **branco**: branca, brancas, branco, brancos
- **criado**: criada, criadas, criadinha, criadinhas, criadinho, criadinhos, criado, criados
- **crioulo**: crioula, crioulas, crioulinha, crioulinhas, crioulinho, crioulinhos, crioulo, crioulos

- **escravo**: escrava, escravas, escravinha, escravinhas, escravinho, escravinhos, escravo, escravos
- **mulato**: mulata, mulatas, mulatinha, mulatinhas, mulatinho, mulatinhos, mulato, mulatos
- **negro**: negra, negras, negrinha, negrinhas, negrinho, negrinhos, negro, negros
- **senhor**: senhor, senhora, senhoras, senhores

Como uma sentença pode conter mais do que uma palavra-alvo é mantida a posição de cada ocorrência. Isso é importante para identificar corretamente as instâncias candidatas de padrões morfo-semânticos.

4.3. Gerar *Embeddings*

Nesta etapa, as sentenças relevantes são submetidas ao BERTimbau pré-treinado na língua portuguesa na sua versão grande ($BERTimbau_{Large}$) para gerar os *embeddings*. O texto de entrada (*input*) do BERTimbau pré-treinado é limitado a 512 *tokens*. Assim, sentenças relevantes são submetidas individualmente ao BERTimbau para não extrapolar este limite enquanto se mantém a informação de contexto de cada sentença. São gerados *embeddings* de componentes textuais em 3 níveis de granularidade: sentenças, janelas dentro de sentenças e palavras individuais. O *embedding* de cada sentença é a média dos *embeddings* de seus *tokens*. Janelas são fragmentos de sentenças centrados em uma palavra-alvo, considerando um certo número (usualmente 1 a 10) de palavras vizinhas à esquerda e à direita, até no máximo o limite da sentença. O *embedding* de uma janela é a concatenação dos *embeddings* dos *tokens* que estão dentro dela. Esses *embeddings* de *tokens* são coletados dos *embeddings* da respectiva sentença. Isso é feito por dois motivos: para obter *embeddings* de janelas que considerem todo o contexto da sentença e para capturar relações da sentença com a janela. Por fim, os *embeddings* de palavras são os *embeddings* dos respectivos *tokens* da sentença. Os *embeddings* de *tokens* são a última camada do BERT, gerando embeddings com 1024 valores (dimensões).

O Algoritmo 1 apresenta na forma de pseudocódigo o programa criado para a tarefa de gerar *embeddings* de sequências de palavras (janelas) que contenham uma determinada palavra-alvo. As sentenças utilizadas nessa tarefa são selecionadas como descrito na Subseção 4.2. Uma lista de registros é criada para armazenar as janelas com seus *embeddings*. Todas as sentenças relevantes são percorridas para gerar os *embeddings*. O registro de cada sentença da lista é composto pelo id da sentença, lista de *tokens*, posição da palavra alvo, a própria palavra alvo e grupo da palavra-alvo. Para cada sentença é realizada a concatenação dos *tokens* utilizando a função *"juncao"*. O texto resultante da junção é passado como parâmetro para a função "*getEmbeddingsText*", para gerar e retornar os seus *embeddings*. Em seguida, a função "*"expandeJanela"* recebe o *token* e sua posição para gerar e retornar os limites inferiores e superiores da janela. Com os limites definidos, é criado o registro da janela contendo: o id da sentença, a sentença, o texto da janela, o comprimento da janela, os *embeddings* de tokens dentro da janela, a palavra-alvo e o grupo desta palavra. O registro é então adicionado à lista de registros. Por fim, é realizada a expansão da janela atual, passando os *tokens* da sentença e os limites da janela acrescentando um *token* de cada lado, se o limite da sentença permitir. Esse processo de expansão ocorre até que não seja mais possível expandir o tamanho da janela dentro da sentença.

Algoritmo 1: Criar lista de janelas

Data: Lista de sentenças("$idSentenca$","$tokens$", "$postagging$", "$palavraAlvo$", "$grupo$", "$posAlvo$") de sentenças relevantes (SR)
Result: Lista de janelas(registro)

```
1.  listaRegistro ← [ ]
2.  foreach index, reg in SR do
3.      idSentenca ← reg["idSentenca"]
4.      tokens ← reg["tokens"]
5.      posAlvo ← reg["posAlvo"]
6.      palavraAlvo ← reg["palavraAlvo"]
7.      grupo ← reg["grupo"]
8.      texto ← juncao(tokens)
9.      embSentenca ← getEmbeddingsText(texto)
10.     tamJanela ← 1
11.     janelaInf, janelaSup, expande ←
          expandeJanela(token, posAlvo, posAlvo)
12.     while expande = True do
13.         embJanela ← embSentenca[janelaInf : janelaSup]
14.         janela ← juncao(tokens[janelaInf : janelaSup])
15.         compJanela ← tamJanela * 2 + 1
16.         registro ← [idSentenca, texto, janela, compJanela,
              embJanela, palavraAlvo, grupo]
17.         listaRegistro.append(registro)
18.         janelaInf, janelaSup, expande ←
              expandeJanela(token, janelaInf, janelaSup)
19.         tamJanela ← tamJanela + 1
```

4.4. Visualização e seleção de embeddings

Nesta etapa, os dados gerados como explicado na seção 4.3 são utilizados para gerar dois arquivos *.tsv* padronizados que servem de entrada para o *Embedding Projector*. Um desses arquivos contém a lista de *embeddings* a visualizar. O outro arquivo contém rótulos (*labels*) para representar características dos respectivos *embeddings* (e.g., obra, autor, ano, movimento literário, classe morfossintática ou classe de sentido de *embedding* de palavra, palavra-alvo de um *embedding* de janela). O *Embedding Projector* escolhe uma cor de uma lista de cores para representar *embeddings* rótulos distintos de uma característica (e.g. autores distintos). Isso possibilita identificar uma característica escolhida de cada ponto na visualização (*embedding)* de acordo com a cor usada para exibí-lo.

O *Embedding Projector* possibilita avaliar a distribuição espacial e a vizinhanças dos *embeddings* nele carregados. Isso pode ser feito usando uma das técnicas de redução de dimensionalidade para verificar se os *embeddings* quando projetados em duas ou três dimensões apresentam grupos bem definidos de pontos, polarização ou não servem para o experimento por terem distribuição muito esparsa, sem formar grupos. Essa visualização dos *embeddings* projetos em duas ou três dimensões pode ser feita por especialistas do domínio. Se o conjunto de dados apresentar alguma propriedade relevante ou desejável,

ele pode ser selecionado e separado para ser utilizado na tarefa na mineração de padrões.

4.5. Calcular distâncias/similaridades

O cálculo das medidas de similaridade e distância é realizado para cada corpus e respectivos *embeddings* selecionados pelos especialistas de domínio. São calculadas a distância (Euclidiana e Manhathan) e a similaridade (cosseno) para todos os pares de *embeddings* de palavras, de janelas e de sentenças. Todas as distâncias entre pares de *embeddings* ficam pré-calculadas e armazenadas para reuso pela próxima e última etapa do processo proposto, a mineração de padrões, a fim de evitar recalculá-las para os mesmos em execuções de algoritmos de mineração de padrões.

O Algoritmo 2 apresenta na forma de pseudocódigo o procedimento criado para realizar a tarefa de calcular distâncias/similaridades entre janelas que contenham uma palavra-alvo. Uma lista de registros é criada para armazenar as medidas entres janelas. Todos os *embeddings* de janelas são percorridas e comparados entre si. O registro de cada *embeddings* de janelas da lista é composto pelo id da sentença (”idSentenca”), sentença, janela, tamanho da janela (”compJanela”), lista de *embeddings* (”embJanela”), palavra-alvo e grupo da palavra-alvo. Para calcular as medidas de distâncias/similaridades é realizada a comparação de todos os *embeddings* de janelas ($n^2 - n$). A comparação dos *embeddings* das janelas é realizada pela função ”*getMeasurementsEmbedding*”que recebe os *embeddings* de duas janelas para gerar e retornar as medidas de comparação. Com as medidas geradas é criado o registro $regcomp_a$ contendo os dados da comparação da janela a com b e o registro $regcomp_b$ da comparação da janela b com a. Cada registro é formado por: id da sentença, sentença, janela, tamanho da janela, palavra-alvo, grupo da palavra-alvo, id da sentença comparada, sentença comparada, tamanho da janela comparada, palavra-alvo comparada, grupo da palavra-alvo comparada e a medida. Por fim, os registros são inseridos na lista de medidas de janelas.

4.6. Mineração de padrões morfo-semânticos

Finalmente, a tarefa de minerar padrões morfo-semânticos em torno de palavras-alvo nas sentenças dos textos pode usar qualquer uma das medidas de distância ou similaridade entre *embeddings* (de palavras, de janelas de texto dentro de sentenças ou de sentenças inteiras) calculadas na etapa anterior do processo proposto (Subseção 4.5). O Algoritmo 3 descreve na forma de pseudocódigo a função criada para minerar padrões morfo-semânticos com base em alguma medida de similaridade entre *embeddings* de janelas de texto dentro de sentenças. Um dicionário é criado para armazenar as ocorrências de janelas. Todas as janelas são percorridas para realizar o agrupamento delimitado pelo valor do *threshold*. Cada registro é formado pelo id da sentença, sentença, janela, comprimento da janela, palavra-alvo, grupos de cada janelas comparada e a medida. Para cada registro é verificado se a medida está acima de um *threshold*. A medida estando acima do *threshold* é gerado um identificador formado pela concatenação do id da sentença, a palavra-alvo e janela. O identificador será a chave única no dicionário. Para cada novo identificador é criado um conjunto vazio no dicionário. Para cada ocorrência de um identificador é adicionado o registro ao conjunto de sua respectiva chave. Finalmente é possível realizar uma contagem de cada item desse dicionário e identificar qual janela apresenta a maior ocorrência.

5. Experimentos e resultados

Esta seção relata os experimentos para avaliar a nossa proposta para minerar padrões morfo-semânticos em textos literários visando apoiar análise de discursos. Primeiramente, a Subseção 5.1 descreve alguns detalhes da implementação do processo proposto. A Subseção 5.2 descreve os conjuntos de dados usados nos experimentos. A Subseção 5.3 apresenta as análises e visualizações dos *embeddings* obtidos do conjunto de dados da obra. Por fim, a Subseção 5.5 apresenta e discute os resultados obtidos pela mineração de padrões morfo-semânticos.

5.1. Implementação

O processo proposto foi implementado na linguagem de programação *Python* versão 3.7.13, sobre notebooks do ambiente de execução Google colaboratory [8]. O uso de notebooks visa facilitar a implementação, demonstração e a avaliação dos resultados obtidos. O ambiente Colaboratory também viabiliza experimentos que requerem computadores de alto desempenho com unidades de processamento gráfico (do inglês, *Graphics Processing Unit* - GPU) e unidades de processamento de tensores (do inglês, *Tensor Processing Unit* - TPU) para serem realizados em tempo hábil. A linguagem Python do ambiente vem pré-configurada facilitando o uso da biblioteca Transformers [Wolf et al. 2020] versão 4.5.1 da Huggingface [9], um provedor de código aberto de tecnologias de PLN que implementa a arquitetura padrão do BERT. Dentre os modelos do BERT pré-treinados, utilizamos um modelo treinado para a língua portuguesa $BERTimbau$[10] [Souza et al. 2020] no tamanho *large*, no formato "*cased*" (com caracteres maiúsculos e minúsculos) disponíveis gratuitamente. Além da ferramenta spaCy [11] versão 3.2.0, com vários recursos para processamento de linguagem natural, incluindo funcionalidades para segmentação dos documentos e a análise sintática das sentenças.

5.2. Conjunto de dados

Utilizamos um conjunto de dados sugerido por especialistas em literatura e linguística, composto por 5 obras literárias de autores brasileiros com temas relacionados escravidão, abolicionismo e questões sociais. Os textos integrais dessas obras estão disponíveis na BLPL [12]. A Tabela 2 apresenta os nomes das obras em ordem alfabética (coluna "Nome"), nome do autor ("Autor"), Ano de Publicação, Movimento Literário e Temática. Note que essas obras foram publicas entre 1869 e 1890, sendo na maioria pertencentes ao movimento literário romantismo e uma delas pertencente ao realismo.

A Tabela 3 apresenta estatísticas sobre a quantidade de sentenças, palavras e tokens encontrado nos textos das obras literárias consideradas, após as tarefas de segmentação de sentenças e geração de *emmbeddings*. Para cada obra é apresentado da quantidade de sentenças ("Qtd. Sentença"), quantidade de palavras por sentença ("Qtd. Palavras"), quantidade de tokens gerados pelo BERT ("Qtd. tokens BERT") e quantidade e percentual de palavras desconhecidas pelo BERT ("Qtd.e % de Palavras desconhecidas"), para as quais o BERT gera ngrams (subpalavras com n caracteres).

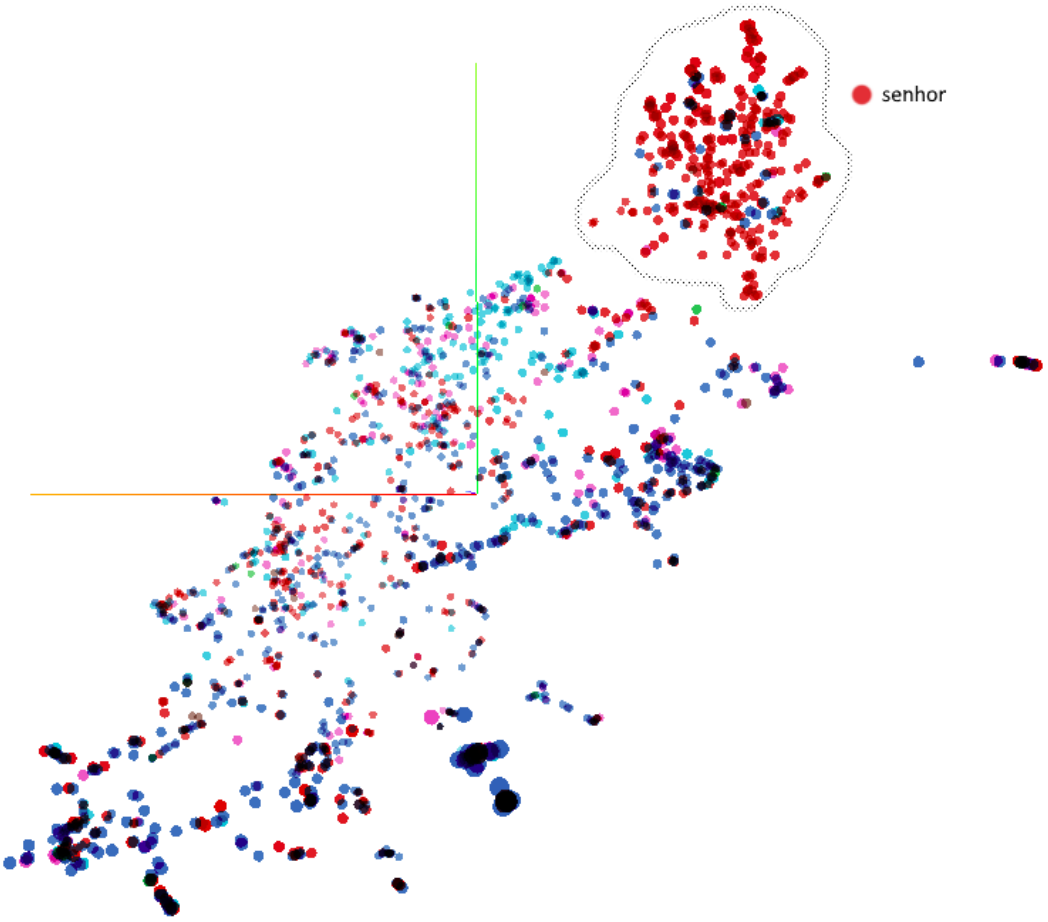
[8] https://colab.research.google.com/notebooks/intro.ipynb
[9] https://huggingface.co/transformers/index.html
[10] https://github.com/neuralmind-ai/portuguese-bert/
[11] https://spacy.io/
[12] https://www.literaturabrasileira.ufsc.br/?locale=pt_BR

5.3. Análise e visualização dos embeddings

Realizamos a segmentação obtendo 6.044 sentenças, posteriormente selecionamos as sentenças relevantes que possuem pelo menos uma palavra alvo, resultando em 1.844 sentenças. Com estas sentenças obtivemos 3 listas: lista dos *embeddings* consolidados das palavras de cada sentença, lista dos *embeddings* consolidados das janelas de tamanho 3 nas sentenças e a lista dos *embeddings* das palavras de cada sentença.

A Figura 5 apresenta a visualização em três dimensões utilizando projeção UMAP com 20 vizinhos (*neighbors*) dos *embeddings* das sentenças consolidados com palavras relevantes da obra *As Vítimas-Algozes*. Nesta figura, cada cor identifica sentenças com uma determinada palavra-alvo. Percebe-se um agrupamento de pontos em vermelho (destacado) no canto superior direito, referente a sentenças que possuem palavras-alvo do grupo ”senhor”. Os grupos de outras palavras-alvo estão em outras cores: marrom para o grupo da palavra-alvo ”branco”, verde para o grupo da palavra-alvo ”criado”, azul-claro para o grupo de ”crioulo”, azul para o grupo o grupo de ”escravo”e cor rosa para o grupo de ”negro”.

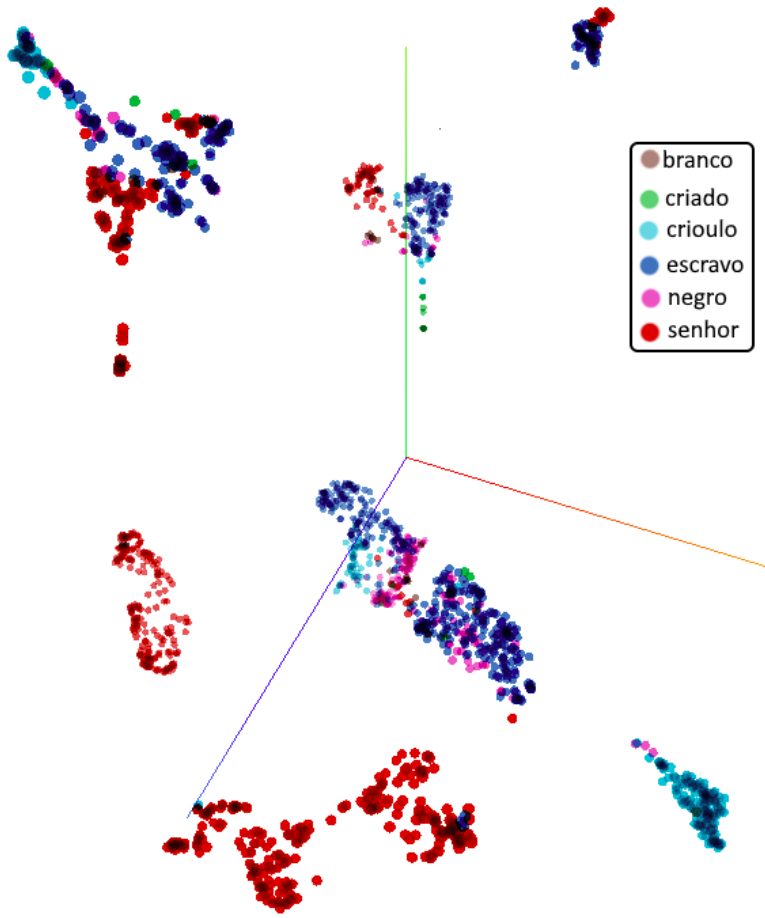
Figura 5. Projeção UMAP 3D de *embeddings* consolidados de sentenças da obra *As Vítimas-Algozes*.



A Figura 6 apresenta a visualização em três dimensões gerada por projeção UMAP de *embeddings* de janelas de palavras de tamanho 3 de sentenças relevantes da obra *As*

Vítimas-Algozes. Nessa figura, podemos identificar 7 grupos bem definidos de pontos. Destacam-se à esquerda dois grupos na cor vermelha referente a janelas com a palavra central ”senhor”. Outros grupos com sobreposições são identificados nas cores azul-claro para o grupo ”crioulo”, marrom o grupo ”branco”, verde o grupo ”criado”, azul o grupo ”escravo”e na cor rosa o grupo da palavra-alvo ”negro”.

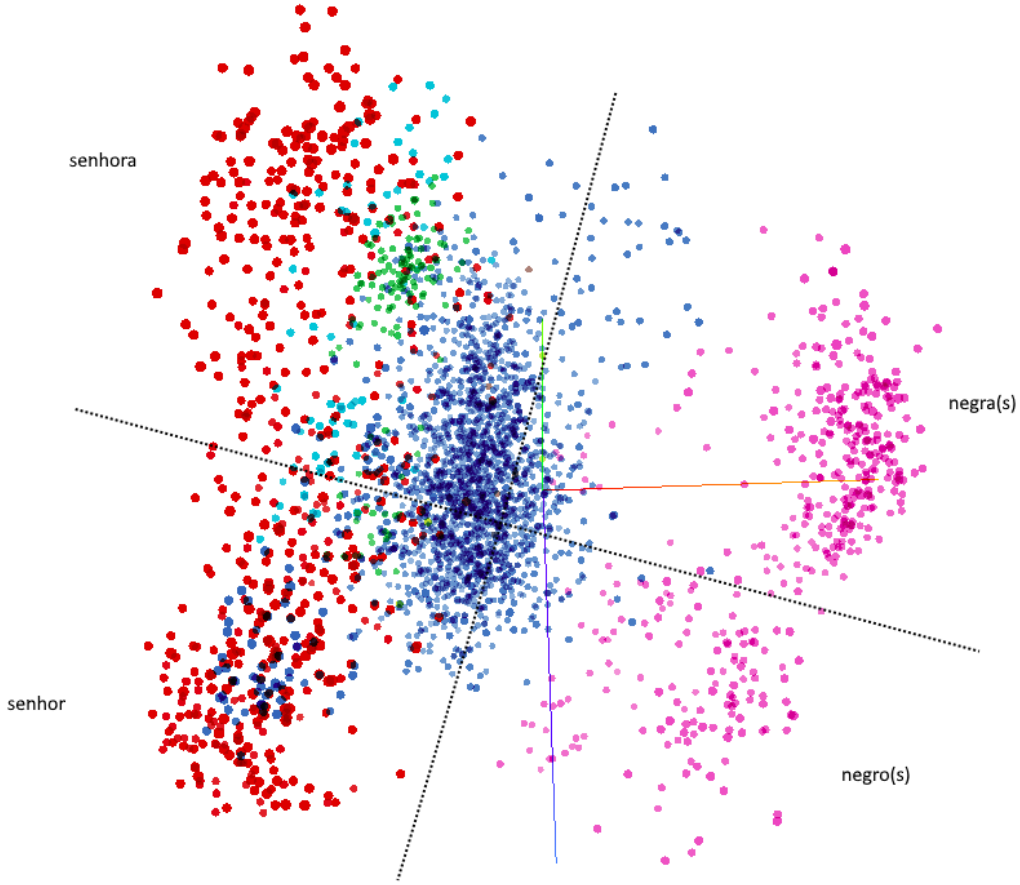
Figura 6. Projeção UMAP de embeddings de janelas com 3 palavras nas sentenças da obra *As Vítimas-Algozes*.



Na visualização dos *embeddings* de palavras das sentenças relevantes, optamos pelo recorte, pois a obra *As Vítimas-Algozes* apresenta um grande número palavras, não sendo possível identificar agrupamentos das palavras-alvo com as demais palavras da obra. Optou-se então por isolar e seleciona os *embeddings* segundo quadrantes para facilitar a visualização de um conjunto menor de pontos. A Figura 7 apresenta a visualização em três dimensões utilizando a técnica PCA do recorte do segundo quadrante dos *embeddings* de palavras das sentenças relevantes da obra *As Vítimas-Algozes*. Com isso identificou-se quatro grupos bem definidos, sendo eles: senhor (cor vermelha), negro (cor rosa), crioulo (verde-claro) e demais palavras (cor azul). Outro ponto importante é a polaridade que foi apresentada, em que temos: polaridade à direita com o grupo ”senhor” e

à esquerda o grupo ”negro”. Além disso, ainda se tem a polaridade de gênero, onde acima temos as palavras do gênero feminino e abaixo as palavras do gênero masculino, destacando esse comportamento para os grupo ”negro” e ”senhor”.

Figura 7. Recorte da projeção PCA dos *embeddings* de palavras da obra *As Vítimas-Algozes*.



5.4. Mineração de padrões morfo-semânticos nos textos

A mineração dos padrões morfo-semânticos nos textos literários usou a similaridade do cosseno entre *embeddings* de janelas de texto centradas em palavras-alvos com limiares (*thresholds*) de $0,8$ e $0,9$ para caracterizar similaridade , conforme utilizado por [Sorato et al. 2020].

Abaixo listamos as 5 sentenças com maior número de ocorrências de palavras-alvo do grupo ”negro”. A primeira tem 3.204 sentenças cuja similaridade com ela é maior ou igual $0,8$.

1. ”Chegados à sala de jantar o crioulo mostrou ao lado direito a porta do aposento de Hermano e de Florinda ; dois dos perversos , o Barbudo e um dos negros colocaram-se aos lados dessa porta : o outro negro recebeu a lanterna e seguiu a Simeão que avançou para a frente e entrou no quarto de dormir de sua senhora .”

2. ”mas , pois que a do Pai-Raiol e de Esméria lhe aproveitava , reputou afortunada a compra que mantinha a consoladora sociedade do negro e da negra que se diziam amar .”
3. ”– Sou negra escrava lançada no campo : animal solto e livre , se eu me desforrasse do desprezo em que meu senhor me abandona , abrindo a porta da minha senzala aos negros meus parceiros e do meu gosto , faria muito bem .”
4. ”Simeão , esquecendo o golpe que recebera , e o sangue que do ombro lhe corria , deixou um dos negros na sala onde estavam os dois cadáveres e com o Barbudo e o outro negro que levava a lanterna , voltou ao quarto da senhora assassinada , arrombou facilmente a gaveta da velha mesa , e apoderando-se de uma grossa chave , foi ao fundo do quarto , arrancou precipitado uma cortina de chita que cobria pequena parte da parede e mostrando em grande vão que havia nesta uma caixa de jacarandá chapeada de ferro , abriu rápido duas fechaduras , e escancarou a caixa que estava cheia de pequenos sacos contendo moedas de ouro e prata.”
5. ”É o santo negro que ajuda os diabos negros !”

A Tabela 4, lista as 5 janelas de sentenças com maior número de ocorrências de palavras-alvo do grupo ”negro”. Estas são janelas das sentenças relevantes (com palavras-alvo) com maior número de outras janelas similares também relevantes. A primeira janela da lista tem 23 janelas similares a ela. A repetição de conteúdos de janelas na terceira e na quarta linhas lista ocorreu porque as janelas ali apresentadas pertencem a sentenças diferentes na obra que possuem *embeddings* contextualizados distinto. As quantidades de janelas similares a elas (com similaridade maior do que $0,8$ limites utilizados por [Sorato et al. 2020]) são 22 e 21, respectivamente.

A Tabela 5, apresenta as 23 janelas de sentenças com *embeddings* mais similares a janela ”O negro conservava-se”, nela é possível entender melhor o padrão minerado.

5.5. Discussão

O conjunto de dados utilizado apresenta um vocabulário rico devido ao tipo de texto analisado, onde o escritor tende a não repetir a mesma frase ou palavra, aproveitando-se dos sinônimos e alternância para transmitir a informação desejada. A segmentação de sentenças (descrita na Subseção 4.1) gerou sentenças grandes, com algumas contendo mais do que 250 tokens, além de sentenças com mais de uma palavra-alvo. Isso pode ter influenciado na mineração de padrões (tarefa descrita na Subseção 4.6), que resultou em baixo número de padrões morfo-semânticos em volta das palavras-alvo encontradas.

O idioma português apresenta um vocabulário muito rico, podendo ter flexões para indicar conjugação verbal, grau do substantivo e outras. Isso possibilita ao autor descrever uma cena de várias formas. Além disso, há a possibilidade de alguma sentença não relevante (i.e., que não contenha palavra-alvo) apresentar um alto grau de similaridade com sentenças relevantes, constituindo um padrão a ser analisado.

Na tarefa de geração de *embeddings* (descrita na Subseção 4.3) houve perda de informação devido ao grande volume de tokens desconhecidos. Talvez se utilizarmos um modelo do BERT treinado com os textos literários do mesmo período das obras analisadas, esta perda de informação seja minimizada, o que pode ter efeito positivo nas comparações do *embeddings* (descrita na Subseção 4.5).

A tarefa de visualizar e selecionar *embeddings* (descrita na Subseção 4.4) possibilitou identificar alguns bons agrupamentos. Principalmente as visualizações de *embeddings* de janelas em volta de algumas palavras-alvo relevantes mostraram agrupamentos interessantes. Por exemplo, como os apresentado na Figura 6, o agrupamento das janelas em torno de palavras-alvo do grupo "senhor"e do grupo "crioulo"são bem definidos. Todavia, o *Embedding Projetor* não permite replicar as visualizações geradas, devido às características do algoritmos que utiliza. O *Embedding Projetor* fornece um meio de salvar uma visualização gerando um *bookmark*. Entretanto, esse *bookmark* é estático e não possibilita quase nenhuma interação.

Finalmente, a configuração de hardware fornecida pelo *Google colaboratory* limitou os experimentos nas tarefas de gerar *embeddings* e minerar padrões, pois foram utilizados somente *embeddings* da última camada do BERT, para que as estruturas criadas nos experimentos não consumissem todos os 12 GB de memória disponibilizados. A ideia inicial era seguir a recomendação do [Devlin et al. 2019] que sugere utilizar a concatenação das 4 últimas camadas do BERT que geraria *embeddings* com $4,096$ valores (dimensões). Isso pode ter comprometido as medidas de similaridade e distância e a mineração de padrões com tais medidas. Outro problema foi tempo de execução de cada experimento. Alguns experimentos demoraram cerca de 30 minutos até 2 horas e 40 minutos para serem executados. Mas o *Google colaboratory* encerra a sessão caso identifique inatividade do notebook por tanto tempo. Por conta destas limitações, os experimentos foram reduzidos a utilizar somente sentenças que apresentassem pelo menos uma palavra-alvo.

6. Conclusões e trabalhos futuros

Neste trabalho, desenvolvemos uma abordagem baseada em PLN para minerar padrões morfo-semânticos em textos literários visando suportar classificação não supervisionada e análise semântica de discursos em torno de tópicos fornecidos, em um estudo de caso na área de literatura. Propusermos um algorítimo que utiliza o modelo chamado $BERTimbau$ uma versão em português do BERT para o processamento de textos literários. O processo se inicia pela segmentação do texto em sentenças, POS-Tagging e lematização, selecionar sentenças relevantes, gerar instâncias candidatas, gerar visualização de *embeddings* de instâncias e, por fim, analisar a distribuição dos *embeddings* das instâncias.

Os resultados experimentais revelaram padrões determinados por similaridades entre sentenças. Sentenças contendo palavras-alvo de um mesmo grupo tendem a ficar próximas. Por exemplo, as sentenças que apresentavam à palavra-alvo "senhor"e "senhora" ficaram próximas quando utilizamos o *Embedding Projector*, e o mesmo fenômeno se repetiu com as visualizações das janelas. Porém, nas visualizações dos *embeddings* das palavras foi possível identificar que o conjunto de dados apresentou polaridade de gênero e de classe social.

Trabalhos futuros incluem: (*i*) aprimorar os algoritmos para mineração de padrões morfo-semânticos baseados em proximidade de *embeddings*, inclusive ao nível de palavras individualmente; (*ii*) treinar modelo de linguagem com textos de obras literárias do mesmo período das obras mineradas; (*iii*) explorar a técnica de janelas deslizantes para tentar gerar melhores *embeddings*; (*iv*) utilizar modelos de *word embeddings* mais recen-

tes, tais como o *BLOOM*[13] e (v) investigar as possibilidades de aplicação da abordagem proposta a textos de outros domínios, tais como textos científicos, jurídicos e de literaturas atuais.

Referências

Araújo-Junior, J. R. d. S. (2021). Mineração de poemas através de técnicas de processamento de linguagem natural.

Braz-Junior, O. d. O. and Fileto, R. (2021). Investigando coerência em postagens de um fórum de dúvidas em ambiente virtual de aprendizagem com o bert. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, pages 749–759, Porto Alegre, RS, Brasil. SBC.

Chen, L., Varoquaux, G., and Suchanek, F. (2021). A lightweight neural model for biomedical entity linking. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, number 35, page 14.

Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Cordeiro, B. C. (2019). *BERT E WORD2VEC: UMA ANALISE INFERENCIAL E COMPUTACIONAL NA CLASSIFICACAO DE TEXTOS COM REDES NEURAIS CONVOLUCIONAIS*. PhD thesis, Universidade Federal do Rio de Janeiro.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Goularte, F. B., Sorato, D., Nassar, S. M., Fileto, R., and Saggion, H. (2020). MSC+: language pattern learning for word sense induction and disambiguation. *Knowl. Based Syst.*, 188.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Leite, J. A., Silva, D. F., Bontcheva, K., and Scarton, C. (2020). Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Muller, P., Braud, C., and Morey, M. (2019). ToNy: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124, Minneapolis, MN. Association for Computational Linguistics.

Ostendorff, M., Ruas, T., Schubotz, M., Rehm, G., and Gipp, B. (2020). Pairwise multiclass document classification for semantic relations between wikipedia articles. *arXiv preprint arXiv:2003.09881*.

[13] `https://huggingface.co/bigscience/bloom`

Porto, W. (2020). Venda de livros digitais cresce 115% em três anos, mostra pesquisa.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sheng, D. and Yuan, J. (2021). An efficient long chinese text sentiment analysis method using bert-based models with bigru. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 192–197. IEEE.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. (2021). Deep learning applications for covid-19. *Journal of Big Data*, 8(1):1–54.

Smilkov, D., Thorat, N., Nicholson, C., Reif, E., Viégas, F. B., and Wattenberg, M. (2016). Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469*.

Sorato, D., Goularte, F. B., and Fileto, R. (2020). Short semantic patterns: A linguistic pattern mining approach for content analysis applied to hate speech. *International Journal on Artificial Intelligence Tools*, 29(02):2040002.

Sorato, D., Goularte, F. B., Nassar, S. M., and Fileto, R. (2016). Análise de métodos e ferramentas para reconhecimento de palavras relevantes em microblogs. In *Anais do XII Simpósio Brasileiro de Sistemas de Informação*, pages 345–352. SBC.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: pretrained bert models for brazilian portuguese. In *Brazilian conference on intelligent systems*, pages 403–417. Springer.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *CoRR*, abs/1905.05950.

Willrich, R., Mittmann, A., Fileto, R., and dos Santos, A. L. (2020). Capture and visualisation of text understanding through semantic annotations and semantic networks for teaching and learning. *J. Inf. Sci.*, 46(4).

Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zhang, Z., Zhao, H., and Wang, R. (2020). Machine reading comprehension: The role of contextualized language models and beyond. *arXiv preprint arXiv:2005.06249*.

Algoritmo 2: Criar lista de medidas entre janelas

Data: Lista de embeddings de janelas ("$idSentenca$", "$sentenca$", "$janela$", "$compJanela$", "$embJanela$", "$palavraAlvo$", "$grupo$") de sentenças relevantes (SR)

Result: Lista de registro

```
1.  listaRegistro ← [ ]
2.  foreach index_a, reg_a in SR do
3.      idSentenca_a ← reg_a["idSentenca"]
4.      sentenca_a ← reg_a["sentenca"]
5.      janela_a ← reg_a["janela"]
6.      compJanela_a ← reg_a["compJanela"]
7.      embJanela_a ← reg_a["embJanela"]
8.      palavraAlvo_a ← reg_a["palavraAlvo"]
9.      grupo_a ← reg_a["grupo"]
10.     foreach index_b, reg_b in SR do
11.         idSentenca_b ← reg_b["idSentenca"]
12.         sentenca_b ← reg_b["sentenca"]
13.         janela_b ← reg_b["janela"]
14.         compJanela_b ← reg_b["compJanela"]
15.         embJanela_b ← reg_b["embJanela"]
16.         palavraAlvo_b ← reg_b["palavraAlvo"]
17.         grupo_b ← reg_b["grupo"]
18.         if idSentenca_a != idSentenca_b OR janela_a != janela_b then
19.             medida ← getMeasurementsEmbedding(embJanela_a, embJanela_b)
20.             regcomp_a ← [idSentenca_a, sentenca_a, janela_a, compJanela_a, palavraAlvo_a, grupo_a,
                            idSentenca_b, sentenca_b, janela_b, compJanela_b, palavraAlvo_b, grupo_b,
                            medida]
21.             regcomp_b ← [idSentenca_b, sentenca_b, janela_b, compJanela_b, palavraAlvo_b, grupo_b,
                            idSentenca_a, sentenca_a, janela_a, compJanela_a, palavraAlvo_a, grupo_a,
                            medida]
22.             listaRegistro.append(regcomp_a)
23.             listaRegistro.append(regcomp_b)
```

Algoritmo 3: Mineração de padrões morfo-semânticos usando *embeddings* de janelas textuais

Data: Lista de medidas de janelas("$idSentenca_a$", "$sentenca_a$", "$janela_a$", "$compJanela_a$", "$palavraAlvo_a$", "$grupo_a$", "$idSentenca_b$", "$sentenca_b$", "$janela_b$", "$compJanela_b$", "$palavraAlvo_b$", "$grupo_b$", "$medida$")

Result: dicionario de ocorrências de padrões semânticos em janelas

```
1.  dicionario ← dict()
2.  foreach index, reg in listaJanelas do
3.      idSentenca ← reg["idSentenca_a"]
4.      palavraAlvo ← reg["palavraAlvo_a"]
5.      janela ← reg["janela_a"]
6.      medida ← reg["medida"]
7.      if medida >= threshold then
8.          identificador = idSentenca + "__" + palavraAlvo + "__" + janela
9.          if (identificador in dicionario) == False then
10.             dicionario[identificador] ← set()
11.         dicionario[identificador].append(reg)
12. foreach index, janela in dicionario do
13.     if (janela in dicionario) == False then
14.         dicionario[janela] ← 1
15.     dicionario[janela] ← dicionario[janela] + 1
```

Tabela 2. Conjunto das obras literárias

Nome	Autor	Ano de Publicação	Movimento Literário	Temática
A Escrava Isaura	Bernardo Guimarães	1875	Romantismo	Abolicionista
As Vítimas-Algozes	Joaquim Manuel de Macedo	1869	Romantismo	Escravidão
Memórias Póstumas de Brás Cubas	Machado de Assis	1880	Romantismo	Retrata a escravidão, as classes sociais, o cientificismo e o positivismo da época
O Cortiço	Aluísio Azevedo	1890	Realismo	Trata das questões sociais e outra que trata das questões individuais e sentimentais
Úrsula	Maria Firmina	1859	Romantismo	Negritude a partir da perspectiva do próprio negro

Tabela 3. Quantidades de sentenças, palavras e tokens nas obras consideradas

Nome	Qtd. Sentenças	Qtd. Palavras	Qtde. tokens BERT	Qtde. e % Palavras desconhecidas
A Escrava Isaura	3.581	6.6348	86.143	5.871 (8,85%)
As Vítimas-Algozes	6.044	121.192	160.451	8.715 (7,19%)
Memórias Póstumas de Brás Cubas	4.678	78.371	103.617	7.220(9,21%)
O Cortiço	6.163	101.582	132,889	8.442 (8,31%)
Úrsula	3.439	56.241	75.574	5.090 (9,05%)

Tabela 4. Lista das 5 janelas de sentenças com maior quantidade de ocorrência similaridade entre os *embeddings*

Janela	Fragmento da sentença	Qtd.
”O negro conservava-se”	[...]” **O negro conservava-se** imóvel e como insensível.”[...]	23
”O negro arrancou-se”	[...]” **O negro arrancou-se** dos braços da crioula, e fitando-a de novo, com olhar imponente de sua vontade, absoluto, imperioso disse ainda, dando à voz tom ameaçador: –”[...]	22
”o negro riu-se”	[...]”Daí a pouco **o negro riu-se**, olhando para o ventre da crioula.”[...]	22
”O negro sacudiu”	[...]” **O negro sacudiu** com a cabeça, e tornou com voz comprimida e alterada:”[...]	21
”O negro riu-se”	[...]” **O negro riu-se** outra vez e disse:”[...]	21

Tabela 5. Lista das 23 janelas com *embeddings* similares à janela "O negro conservava-se"

Janela	Fragmento de Sentença
"O negro ficou"	[...]"O negro ficou impávido; mas franziu as sobrancelhas."[...]
"O negro olhou"	[...]"O negro olhou suspeitoso, mas soberbo para a crioula, e viu a lascívia abrasando-lhe o rosto."[...]
"O negro apertou-lhe"	[...]"O negro apertou-lhe a mão e sentou-se à porta da senzala: a crioula imitou-o sentando-se a seu lado."[...]
"o negro ,"	[...]"como acabava de dizer o negro, muito mais bonita e elegante do que sua senhora."[...]
"O negro insistia"	[...]"O negro insistia ainda no conselho ou ordem que dera a Esméria, a qual continuava a fingir-se hesitante."[...]
"o negro fixou"	[...]"mas o negro fixou os olhos na ninhada de pintainhos, como se os quisesse absorver nas órbitas."[...]
"o negro ,"	[...]"– Dantes era melhor – disse o negro, sossegado."[...]
"O escravo preferia"	[...]"O escravo preferia ver talhar-se uma mortalha."[...]
"o negro que"	[...]"Esméria considerava, contemplava ansiosa o negro que, imóvel e de olhos fitos, mirava a ninhada infeliz."[...]
"O escravo não"	[...]"O escravo não precisava ser seduzido para encarregar-se da comissão: não tinha em estima o recato de sua"[...]
"o negro ;"	[...]"Teresa lembrou-se da impressão repulsiva que experimentara vendo o negro;"[...]
"O negro arrancou-se"	[...]"O negro arrancou-se dos braços da crioula, e fitando-a de novo, com olhar imponente de sua vontade,"[...]
"O negro deixava"	[...]"O negro deixava indiferentemente à mercê de Esméria a vida do senhor, a quem não segurava, nem empurrava."[...]
"O negro riu-se"	[...]"O negro riu-se outra vez e disse:"[...]
"O negro sacudiu"	[...]"O negro sacudiu com a cabeça, e tornou com voz comprimida e alterada:"[...]
"O negro afagou"	[...]"O negro afagou a crioula como costumava, insinuando-se possuído de paixão cada vez mais violenta."[...]
"o negro riu-se"	[...]"Daí a pouco o negro riu-se, olhando para o ventre da crioula."[...]
"O negro ria-se"	[...]"O negro ria-se de modo a causar pavor;"[...]
"o negro tornou-se"	[...]"Entretanto e por isso mesmo que o segredo desaparecera, **o negro tornou-se** mais exigente e aos domingos e dias santificados reclamava com "[...]
"O negro encolheu"	[...]"O negro encolheu os ombros."[...]
"O negro riu-se"	[...]"O negro riu-se;"[...]
"O negro pôs-se"	[...]"O negro pôs-se a rir com o seu medonho riso: ele sabia que a crioula não era menos devassa que dantes."[...]
"O negro pareceu"	[...]"O negro pareceu indignado."[...]