UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Nathan Cezar Cardoso

Mineração de padrões morfo-semânticos em textos clínicos

Florianópolis
2022

Nathan Cezar Cardoso

Mineração de padrões morfo-semânticos em textos clínicos

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciência da Computação.
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.

Cardoso, Nathan Cezar
Mineração de padrões morfo-semânticos em textos clínicos / Nathan Cezar Cardoso ; orientador, Osmar de Oliveira Braz Junior, coorientador, Renato Fileto, 2022.
73 p.

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

Inclui referências.

1. Ciências da Computação. 2. Mineração de textos. 3. Processamento de Linguagem Natural. 4. Embeddings. 5. Textos Clínicos. I. Braz Junior, Osmar de Oliveira. II. Fileto, Renato. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Nathan Cezar Cardoso

Mineração de padrões morfo-semânticos em textos clínicos

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

Florianópolis, 2022.

Prof. Jean Everson Martina, 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. Elder Rizzon Santos, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Jônata Tyska Carvalho, Dr.
Avaliador
Universidade Federal de Santa Catarina

À minha amada esposa Cristiane pela paciência, apoio absoluto e incentivo diário. Este trabalho não existiria sem você.
A meus pais, Luiz e Dulci por trabalharem muito para dar aos filhos condições e estímulo para estudar.

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 compartilhados.

Ao meu amigo, Rafael Claumann Bernardes por todos os trabalho em grupo e por todas às vezes que me convenceu a não desistir.

Por fim, agradeço a Deus. Espero estar fazendo jus à segunda chance.

“Se não puder voar, corra. Se não puder correr, ande. Se não puder andar,
rasteje, mas continue em frente de qualquer jeito.”
(Martin Luther King Jr.)

RESUMO

Atualmente, enormes volumes de textos de diversos domínios (microblogs, notícias, artigos, prontuários médicos, etc.) têm sido coletados diariamente em plataformas digitais. Várias ferramentas para Processamento de Linguagem Natural (PLN), mineração de textos e ciência de dados permitem extrair informação, analisar e classificar certos textos conforme os seus conteúdos. Recentemente, técnicas de *embedding* de texto, principalmente *embeddings* contextualizados, têm possibilitado ganhos de desempenho em diversas tarefas de Processamento de Linguagem Natural (PLN). Nosso grupo de pesquisa tem investigado a aplicação de tais recursos na mineração de padrões morfo-semânticos em textos, visando extração e análise de informação. Tais padrões têm se mostrado úteis em tarefas como análise de discurso, desambiguação do sentido de palavras e classificação de textos, usando métodos não-supervisionados, que dispensam grandes volumes de dados rotulados para treinamento, possibilitam certa explicabilidade e flexibilidade, por exemplo no detalhamento de categorias de classificação. Este trabalho desenvolveu e avaliou métodos e algoritmos baseados em PLN e *embeddings* contextualizados para minerar eficientemente padrões morfo-semânticos em textos clínicos (inseridos por profissionais de saúde, como médicos, nos prontuários de pacientes em atendimentos), com o intuito de automatizar a classificação e a triagem desses textos e possibilitar análises de seus conteúdos com métodos inovadores. Os textos clínicos e exemplos dos padrões a serem minerados foram fornecidos por uma empresa que presta serviços a operadoras de planos de saúde, com intermediação de um mestrando a ela ligado. Foram gerados *embeddings* do BERT pré-treinados para a língua portuguesa (*BERTimbau*), assim como classes morfossintáticas e reconhecimento de entidades (e.g., medicamentos, doenças, especialidades médicas) de acordo com terminologia específica da área médica para calcular similaridade e/ou determinar casamento na mineração dos padrões nos documentos. Os embeddings gerados foram utilizados em experimentos de visualização e agrupamento a fim de selecionar conjuntos de dados ao redor das entidades reconhecidas mencionadas nos textos clínicos. Com o uso dos algoritmos desenvolvidos e as visualizações geradas foi possível concluir que o modelo do BERT utilizado usa o contexto dos documentos para gerar os *embeddings* dos medicamentos próximos aos *embeddings* de outras palavras mencionadas nos mesmos contextos textuais, tais como doenças tratadas com os respectivos medicamentos. Isso não permite discriminar medicamentos e doenças, por exemplo, em grupos distintos de *embeddings*.

Palavras-chave: Mineração de textos. Classificação não-supervisionada de textos. Processamento de Linguagem Natural. *Embeddings*. Reconhecimento de entidades. Textos Clínicos.

ABSTRACT

Currently, huge volumes of texts from different domains (microblogs, news, articles, medical records, etc.) have been collected daily on digital platforms. Various tools for Natural Language Processing (NLP), text mining, and data science allow extracting information and analyzing and classifying certain texts according to their contents. Recently, text embedding techniques, mainly contextualized *embeddings*, have enabled performance gains in several NLP tasks. Our research group has investigated the application of such resources in the mining of morpho-semantic patterns in texts, aiming at extracting and analyzing information. Such patterns are useful in tasks such as discourse analysis, disambiguation of the meaning of words, and classification of texts, using unsupervised methods, which do not require large volumes of labeled data for training, allowing some explainability, and flexibility, for example in detailing classification categories. This work aims to develop and evaluate methods based on contextualized NLP and *embeddings* to efficiently mine morpho-semantic patterns in clinical texts (inserted by health professionals, such as doctors, in the records of patients in attendance), intending to automate the classification and sorting of these texts and enable an analysis of their contents with innovative methods. The clinical texts and examples of the standards to be mined have been provided by a company that provides services to health plan operators, with the intermediation of a master's student linked to it. It is intended to use pre-trained BERT *embeddings* for the Portuguese language, as well as morphosyntactic classes and entity recognition (e.g., drugs, diseases, medical specialties) according to specific medical terminology to calculate similarity and/or determine matching by mining the patterns in the documents. The generated embeddings were used in visualization and clustering experiments in order to select datasets around recognized entities mentioned in clinical texts. With the use of the developed algorithms and the generated visualizations, it was possible to conclude that the BERT model used uses the context of the documents to generate the embeddings of the drugs close to the embeddings of other words mentioned in the same textual contexts, such as diseases treated with the respective drugs . This does not allow discriminating drugs and diseases, for example, in distinct groups of embeddings.

Keywords: Text mining. Unsupervised classification of texts. Natural Language Processing. Embeddings. Entity recognition. Clinical Texts.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
BERT	*Bidirectional Encoder Representations from Transformers*
CFM	Conselho Federal de Medicina
CSV	*Comma-separated Values*
DeCS	Descritores em Ciência da Saúde
DL	*Deep Learning -*
GloVe	*Global Vectors for Word Representation*
GPU	*Graphics Processing Unit*
MCL	Modelo Contextualizado de Linguagem
ML	*Machine Learning*
NER	*Named Entities Recognition -*
OCR	*Optical Character Recognition*
OOV	*Out-Of-Vocabulary*
PCA	*Principal Component Analysis*
PLN	Processamento de Linguagem Natural
POPE	Prontuário Orientado por Problemas e Evidências
POR	*Problem-Oriented Record*
POS	*Part-Of-Speech*
QA	*Question Answering*
ROE	*Return On Equity*
SNE	*t-Distributed Stochastic Neighbor Embedding*
SPARQL	*SPARQL Protocol and RDF Query Language*
SSP	*Short Semantic Pattern*
TFIDF	*Term Frequency–Inverse Document Frequency*
TM	*Text Mining -*

TPU *Tensor Processing Unit*

UMAP *Uniform Manifold Approximation and Projection for Dimension Reduction*
UMLS *Unified Medical Language System*
URI *Uniform Resource Identifier*

SUMÁRIO

1 INTRODUÇÃO

O PLN está atualmente conceituado na ciência da computação. Novas aplicações aparecem a todo momento, auxiliando, por exemplo, no entendimento da experiência e satisfação do cliente (PIRIS; GAY, 2021) ou na previsão de resultados judiciais (BERTALAN; RUIZ, 2020). Tarefas de PLN como classificação morfossintática de palavras (do inglês, *Part-Of-Speech* (POS)-*Tagging*), juntamente com *embeddings* de palavras e aprendizado de máquina, permitem também detectar e analisar discursos de ódio em textos de mídias sociais (SORATO; GOULARTE; FILETO, 2020; LEITE et al., 2020), entre várias outras aplicações.

No contexto de saúde, o uso de técnicas e ferramentas de PLN em aplicações também vem aumentando consideravelmente nos últimos anos, sobretudo em resposta à pandemia do COVID-19. Abordagens atuais utilizam técnicas e ferramentas de PLN e mineração de textos (do inglês, *Text Mining* - (TM)) em diferentes áreas da medicina. Por exemplo, Souza e Felipe (2021) utilizam reconhecimento de entidades nomeadas (do inglês, *Named Entities Recognition* - (NER)) (LI et al., 2022) para extração de termos técnicos de anamneses (registro completo da história clínica de um paciente) de prontuários eletrônicos do domínio da ginecologia, enquanto Ridgway et al. (2021) usam PLN em anotações clínicas para detectar doenças mentais e uso de substâncias entre pessoas vivendo com HIV e Sorin et al. (2020) utilizam o PLN e aprendizado profundo (do inglês, *Deep Learning* - (DL)) na área da radiologia. Todavia, ainda há vários desafios em aberto envolvendo PLN no domínio da saúde.

1.1 DESCRIÇÃO DO PROBLEMA

Com o aumento dos tele-atendimentos e extensa digitalização dos dados clínicos (colhidos em atendimentos), cresce cada vez mais a necessidade de se minerar esses dados. Dados (semi)-estruturados, informação e conhecimento extraídos de dados clínicos colhidos na forma de textos em linguagem natural podem auxiliar e aprimorar sistemas de vigilância de saúde (auxiliando na análise de tendências, rápida identificação de doenças e surtos, além de fatores de risco). Isso pode contribuir para a criação de melhores estratégias de prevenção de doenças e resposta às tendências detectadas ou mesmo crises, entre outras possibilidades.

O problema tratado neste trabalho é minerar certos padrões de linguagem, a que denominamos morfo-semânticos, em textos clínicos. Um padrão morfo-semântico (GOULARTE et al., 2020) é caracterizado pela presença de palavras com classes gramaticais equivalentes ou compatíveis e sentidos similares em diferentes instâncias de textos (e.g., sentenças, documentos curtos). Para exemplificar, considere as seguintes sentenças obtidas de textos clínicos processados neste trabalho:

1. *"**Oriento**^{V,O} sobre aumento^{S,A} da ingesta hidrica^{SA,IH}."*

2. *"**Prescrevo**^{V,P} Cetoprofeno^{S,M} e descolonização da pele com Triclosan^{S,M}, manter Mometasona^{S,M}; "*

3. *"**Prescrevo**^{V,P} Liposic^{S,M} 5x ao dia e **buscar**^{V,B} atendimento com Oftalmo^{S,E} caso sintomas persistirem."*

4. *"Oriento **buscar**^{V,B} atendimento com **Dermatologista**^{S,E} caso não haja melhora; "*

Nessas sentenças diferentes verbos determinam parte fundamental da semântica de cada discurso: orientação (V,O), prescrição (V,P) e orientação de busca (V,B) por (encaminhamento a) atendimento especializado. Convém observar que verbos distintos, mas com sentido análogo, podem determinar semântica análoga, em diferentes instâncias de texto. Por exemplo, ao invés de *"oriento/prescrevo ... buscar"* pode ser utilizado *"encaminho"*, para indicar o mesmo sentido geral de encaminhamento. Além disso, embora toda recomendação médica deva ser clara e inequívoca, variações léxicas, sintáticas e erros ortográficos, como da palavra *hidrica*^{SA,IH} (mantida incorreta propositalmente) na sentença 1, podem dificultar a extração da informação ou seu entendimento.

Analogamente, substantivos referindo-se a entidades denotam o que está sendo indicado ao paciente. Ferramentas de PLN já disponíveis podem ser usadas para efetuar não apenas a classificação morfossintática de palavras (POS-*Tagging*) para identificar verbos, substantivos, etc. Elas podem ser usadas também para o reconhecimento de entidades nomeadas (NER) e classificá-las em categorias como alimentos, medicamentos e especialidades médicas. Por exemplo, na sentença 3 há a menção a duas entidades, *Liposic*^{S,M} e *Oftalmo*^{S,E}, que referenciam um medicamento e uma especialidade médica, respectivamente.

Uma ocorrência de verbo no texto pode estar associada a várias entidades, sendo as palavras que denotam tanto o verbo quanto as entidades, compatíveis em termos de classe morfossintática (verbo com verbo, substantivo com substantivo) e sentido (similar ou referente à mesma categoria de entidades) com outra(s) palavras em trechos distintos de textos. Por isso denominamos tais padrões morfo-semânticos. Todavia, vale salientar que para um conjunto de trechos de texto serem instâncias do mesmo padrão morfo-semântico, não precisa necessariamente haver correspondência biunívoca entre suas palavras. Basta haver alguns casamentos de palavras, em termos das respectivas classes morfo-sintáticas e sentidos. Por exemplo, a sentença 2 e a primeira oração da sentença 3, podem ser consideradas instâncias de um mesmo padrão morfo-semânticos, pois ambas fazem prescrição de medicamento(s). Por outro lado, a segunda oração da sentença conjugada 3 e a sentença 4 são instâncias de outro padrão, pois ambas fazem encaminhamento a especialidade médica.

Um padrão morfo-semântico é portanto determinado pelo alinhamento de palavras em trechos distintos de textos (instâncias do padrão). Tal alinhamento pode ser aferido pela compatibilidade morfo-sintática e pela similaridade semântica entre palavras. Os significados dos verbos podem indicar padrões gerais (orientação, prescrição, encaminhamento), enquanto as ca-

tegorias das entidades, usualmente denotadas por substantivos, podem refinar os padrões (prescrição de certo medicamento, encaminhamento a atendimento em certa especialidade, etc.).

1.2 OBJETIVOS

O objetivo geral deste trabalho é minerar padrões morfo-semânticos em textos clínicos, visando classificação e triagem não supervisionada desses textos de acordo com tais padrões e análises de seu conteúdo. Para alcançá-lo, será necessário atingir os objetivos específicos abaixo relacionados.

- Compreender e se habilitar a utilizar métodos, técnicas e ferramentas de PLN, para efetuar tarefas como normalização de texto, POS-*Tagging*, NER e cálculos de medidas de distância e similaridade entre *embeddings* contextualizados de palavras, visando aplicação na mineração de padrões morfo-semânticos em textos clínicos.

- Desenvolver e avaliar algoritmos baseados em tais tecnologias para mineração eficiente de padrões morfo-semânticos em textos clínicos.

- Avaliar qualitativa e quantitativamente os algoritmos propostos, em termos da distribuição das instâncias dos padrões minerados em coleções de documentos, custos computacionais e, na medida das possibilidades, medidas de qualidade dos resultados frente a regras ouro.

1.3 METODOLOGIA

Inicialmente, foram realizados estudos sobre o estado da arte em tarefas e ferramentas de PLN que podem ser usadas na preparação dos textos clínicos a serem minerados, nos próprios algoritmos de mineração de padrões morfo-semânticos e na avaliação dos seus resultados. Tais tarefas incluiram normalização de texto (tokenização, stemming, etc.), classificação morfossintática (POS -*Tagging*), reconhecimento de entidades nomeadas (NER) e cálculo de similaridade entre *embeddings* de palavras. Posteriormente, foram 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-semânticos, incluindo, entre outros, funções de similaridade ou distância semântica e patamares (*thresholds*) a serem utilizadas na mineração.

Existem muitos estudos sobre o uso de NER para área da medicina na língua inglesa (WANG et al., 2020; GLIGIC et al., 2020). Entretanto, a mineração de padrões e mesmo NER em textos clínicos em português ainda não foram devidamente tratados na literatura considerando os atuais *embeddings* contextualizados. Experimentos preliminares de mineração de padrões morfo-semânticos em pequenos conjuntos de textos clínicos usando somente *embeddings*, incluindo os do *BioBERT pt* (LEE et al., 2019; SCHNEIDER et al., 2020), que foi treinado para

textos médicos, sugerem a necessidade de efetuar NER para identificar entidades como medicamentos e especialidades da área de saúde. Todavia, carência de soluções adequadas para NER em textos clínicos em língua portuguesa pode gerar dificuldades para este trabalho. Assim, estamos atualmente pesquisando e selecionando possíveis soluções que estejam disponíveis e possam ser adaptadas para as nossas necessidades.

Para resolver o problema de minerar automaticamente padrões morfo-semânticos podem ser utilizadas diversas alternativas de medidas de similaridade e outros critérios para determinar compatibilidade de palavras. Neste trabalho exploramos *embeddings* contextualizados de palavras e categorias de entidades identificadas mediante aplicação de NER para determinar 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 existentes nos textos clínicos escritos por profissionais da saúde durante os atendimentos médicos. Descrições desses artefatos são apresentadas em detalhes nos capítulos 2, 3 e 4.

Finalmente, foi 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. Os custos computacionais foram avaliados em termos de tempo de processamento e uso de memória.

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 em textos clínicos usando *embeddings* contextualizados. Finalmente o Capítulo 5 delineia o plano de experimentos para avaliar a proposta e reporta experimentos iniciais.

2 FUNDAMENTOS

Este capítulo descreve brevemente os principais conceitos e técnicas usados neste trabalho para a mineração de padrões morfo-semânticos em textos clínicos. Primeiramente, a seção 2.1 fornece uma visão geral das características de textos clínicos e do padrão (SOAP), o qual inclui um protocolo utilizado para coletar dados clínicos em forma textual e um formato para representá-los e armazená-los. Posteriormente, a seção 2.2 descreve as tarefas de Processamento de Linguagem Natural (PLN) usadas nas soluções investigadas neste trabalho. Finalmente, a seção 2.3 apresenta uma breve introdução a *embeddings*, ao modelo contextualizado de linguagem BERT, empregado para gerar os *embeddings* usados neste trabalho, e ao cálculo de funções de distância e similaridade entre *embeddings*.

2.1 TEXTOS CLÍNICOS

Os prontuários são documentos legais que devem conter todos os dados assistenciais prestados aos pacientes, sejam registros de internação ou consultas em consultório, tornando-se um documento importante para a integração do cuidado. Além disso, apresenta informações sociais, demográficas e socioeconômicas. O artigo 1º da Resolução 1.638/2002, de 9 de agosto de 2002, do Conselho Federal de Medicina (CFM) define prontuário como:

> [. . .] documento único constituído de um conjunto de informações, sinais e imagens registradas, geradas a partir de fatos, acontecimentos e situações sobre a saúde do paciente e a assistência a ele prestada, de caráter legal, sigiloso e científico, que possibilita a comunicação entre membros da equipe multiprofissional e a continuidade da assistência prestada ao indivíduo.

Para Pinto e Sales (2017), é uma memória escrita sobre os dados clínicos e não clínicos do enfermo. Eles contêm uma história que serve como elo de comunicação entre as diferentes equipes da instituição e com os pacientes. Portanto, para resumir o entendimento dessas definições, o prontuário é escrito de forma colaborativa, documentando os aspectos físicos, psicológicos e sociais do paciente (GALVÃO; RICARTE, 2011), mesmo levando em conta a decisão médica como provedora do diagnóstico clínico.

Os dados clínicos são considerados informações para monitorar a saúde do indivíduo, obtidas conforme o paciente é observado: exames de imagem e laboratoriais, histórico médico, evolução, prescrição, sinais vitais e avaliação de risco e resumo de alta. Por outro lado, as informações não clínicas estão relacionadas a questões administrativas, não as condições de saúde. São dados relacionados às atividades nutricionais, farmácia, manutenção, aplicação de protocolo e materiais consumidos (PINTO; SALES, 2017).

No cotidiano de uma organização, os prontuários são a base para a coleta de protocolos de saúde em três áreas, corroborando ações judiciais e fonte de pesquisa no meio acadêmico. Conforme o Capítulo X da Resolução CFM 2056/2013, os prontuários devem conter: anamnese

e exame físico, folhas de prescrição e de evolução exclusiva para médicos e enfermeiros, folhas de assentamento evolutivo comum para os demais profissionais que intervenham na assistência. Como tal, é considerado um registro complexo que contém informações ou evidências quanto ao modo de produção, conteúdo, organização e disponibilização. Sendo registrada no papel ou de maneira eletrônica, exige planejamento organizacional, colaboração e trabalho de longo prazo entre gestão das unidades de saúde, profissionais de saúde e da informação para sistematizar dados em prontuários (GALVÃO; RICARTE, 2011).

Profissionais de saúde debatem sobre como deve ser organizado um prontuário médico antes mesmo da criação dos sistemas informatizados. Na década de 1960 Weed (1968) propôs um modelo de prontuário que é atualmente adotado em diversos centros de saúde de todo o mundo (LOPES, 2020). O modelo de prontuário proposto por Weed foi denominado Prontuário Orientado por Problemas e Evidências (POPE) (do inglês, *Problem-Oriented Record* - POR) e tem como pontos mais relevantes a Lista de Problemas e as Notas de Evolução no Modelo **SOAP**. Cada letra da sigla do modelo SOAP se refere a um dos quatro aspectos fundamentais das notas de evolução: **dados subjetivos (S), dados objetivos (O), avaliação (A) e planos (P)** (LOPES, 2020). A Figura 1 ilustra um exemplo de campo **SOAP** preenchido por um profissional da saúde. Na figura, no campo subjetivo (S) o profissional registra os relatos do paciente documentando os sinais e sintomas mencionados, descrevendo o motivo que o trouxe à consulta. No campo objetivo (O), é feito o registro dos dados observáveis e mensuráveis avaliados pelo profissional da saúde, abrangendo o exame clínico e os exames complementares. No campo avaliação (A) o profissional descreve a impressão/interpretação que infere a partir das queixas subjetivas (S) do paciente e dos achados da parte objetiva (O). Neste campo é descrita a lista de problemas da consulta atual. Por último, no campo plano (P) é elaborada uma proposta de abordagem planejada para os problemas levantados. Esta proposta pode compreender medicações prescritas, solicitações de exames complementares, orientações realizadas, encaminhamentos e pendências para o próximo atendimento.

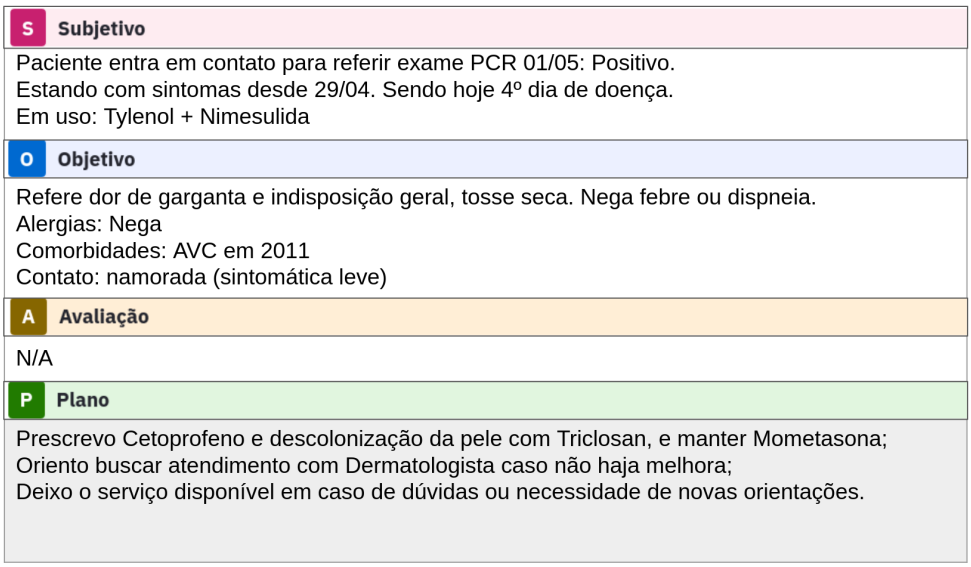
Portanto, a concentração de informação reunida por profissionais de saúde neste documento pode ser usada para formular uma síntese automatizada de dados úteis com base nas necessidades desses profissionais e da gestão das unidades de saúde. Desta forma, apoiando uma melhor tomada de decisão para resolver os casos.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

Segundo Wang et al. (2020) o Processamento de Linguagem Natural (PLN) refere-se à capacidade das máquinas de entender e explicar a maneira como os humanos escrevem e falam. Envolve o estudo de várias teorias e métodos que podem realizar uma comunicação eficaz entre humanos e computadores em linguagem natural e é uma direção importante no campo da inteligência artificial.

Para processar um texto escrito em uma linguagem desenvolvida por humanos, é necessário seguir uma série de etapas iterativas envolvendo o PLN. Essas etapas incluem pré-

Figura 1 – Exemplo de prontuário no padrão SOAP preenchido por um profissional da saúde



Fonte: o autor.

processamento dos dados; representação de palavras como vetores numéricos; treinamento de um modelo de ML usando a representação da etapa anterior; avaliação do modelo. No decorrer desta sessão essas etapas serão aprofundadas.

2.2.1 Pré-processamento

O PLN pode se tornar uma tarefa árdua de realizar se o formato dos dados não for padronizado. Nesse sentido, Ly, Uthayasooriyar e Wang (2020) propõem alguns passos que podem ser realizados nos textos para tornar o processamento de texto mais simples e menos impreciso:

- Carregar o texto: carregar o texto na memória, corretamente, lidando com a codificação desejada;
- *Lowercasing*: operação de converter todas as letras para minúsculas, isso evita que a mesma palavra seja identificada como duas devido às letrar maiúsculas;
- Remoção de *stopwords*: remover palavras menos significativas como preposições, conjunções, etc. porque elas não carregam nenhum significado e não são uteis para a análise;
- Processamento de caracteres especiais: dependendo do contexto, caracteres especiais como interrogações ou exclamações são importantes para o processamento posterior. Contudo, é necessário remover duplicidades ou caracteres indesejados;
- Tokenização: dividir o texto em um conjunto de unidades menores (tokens). Essas unidades podem ser sentenças ou até mesmo palavras.

No geral, o processo de tokenização envolve algumas etapas mais complexas. Ao lidar com palavras, existem processos usados para reduzir o número de palavras no vocabulário que

serão reconhecidas pelo programa posteriormente. Um destes processos se chama *Stemming* e seu objetivo é reduzir as palavras à sua forma mais simples, isto é, ao seu radical. Por exemplo, as palavras "ferrugem" e "ferreiro" seriam transformadas em "ferr", sendo "ugem" e "eiro" seus sufixos que foram removidos.

Outro processo utilizado na tokenização é o *Lemmatization*, cujo objetivo é também converter palavras em sua forma mais básica. Neste processo é necessário analisar morfologicamente as palavras para identificar suas categorias gramaticais e fazer as simplificações necessárias. Por exemplo, as palavras "tiver", "tenho" e "tinha" são todas formas do mesmo lema "ter".

Embora ambos os processos anteriores exijam conhecimento do idioma, há uma abordagem mais geral, a tokenização WordPiece (WU et al., 2016). Nesta abordagem, as palavras são divididas em partes menores e o modelo de ML é usado para rotular palavras no texto mais eficientemente. Por exemplo, a palavra "estudar" pode ser dividida em "estud" e "ar" no processo de treinamento. Quando for solicitado dividir uma nova palavra como "estudando", ele poderá inferir as partes "estud" e "ando" mesmo sem ter visto a palavra original antes.

Por fim, o processo de tokenização *N-grams* é um processo que considera o contexto das palavras. Para isso, são construídas palavras considerando as n-1 palavras anteriores. Por exemplo, na sentença "Não foi nada mal.", em um modelo 1-gram as palavras "nada" e "mal" seriam separadas, mas em um modelo 2-gram a palavra seria "nada mal", melhorando a identificação do sentido da sentença.

2.2.2 Análise morfológica de palavras

As explicações e definições apresentadas a seguir referem-se à morfologia de palavras e à sintaxe de textos. Elas estão de acordo com a gramática da língua portuguesa e podem requerer adaptações ao considerar outras línguas.

A morfologia trata da formação e classificação das palavras, as quais podem ser vistas como unidades básicas da construção linguística. As palavras são classificadas em categorias morfossintáticas de acordo com sua função. As principais categorias morfossintáticas de palavras na língua portuguesa são: substantivo, artigo, adjetivo, numeral, pronome, verbo, advérbio, preposição, conjunção e interjeição. Substantivo e verbo são as classes de palavras mais relevantes para fins de extração de informação, pois costumam denotar as bases semânticas das sentenças (CEGALLA, 2008). Uma mesma palavra pode pertencer a mais de uma categoria, dependendo do contexto onde ela aparece (posição na estrutura sintática do texto, palavras vizinhas, etc.).

A tarefa de PLN responsável por associar automaticamente cada palavra em um texto à sua classe morfossintática é usualmente denotada pelo termo inglês POS-*Tagging*. Atualmente, diversos kits de ferramentas de PLN de código aberto, tais como spaCy[1] e Stanza[2], oferecem

[1] https://spacy.io/

[2] https://stanfordnlp.github.io/stanza/

métodos variados para realizar POS-*Tagging*, os quais podem muitas vezes ser customizados e evocados a partir de linguagens de programação como Python (QI et al., 2020). A Tabela 1 apresenta as classes morfossintático utilizadas pela ferramenta spaCy.

Tabela 1 – 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.

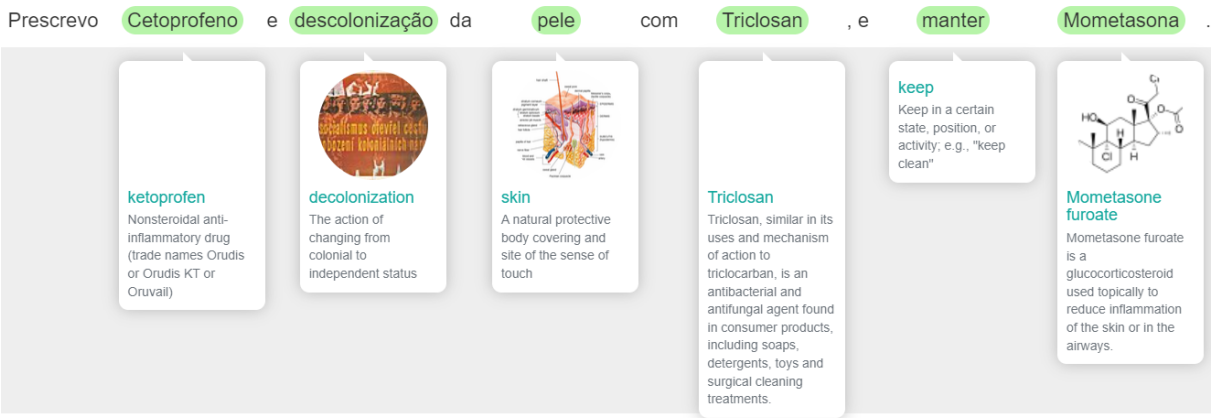
As palavras, tanto na expressão escrita como na oral, são agrupadas e ordenadas em frases e orações, sendo as frases também denominadas sentenças. É através da sentença que se alcança o objetivo do discurso produzido pela atividade linguística: a comunicação com o receptor da mensagem (WANG; GUO, 2014). Na língua portuguesa, a análise sintática examina a estrutura de um texto, divide-o em sentenças, orações e seus subcomponentes, além de classificá-los (CEGALLA, 2008). Na Seção 2.2 definimos os conceitos e tarefas utilizadas para processar linguagem natural.

2.2.3 Reconhecimento de Entidades Nomeadas

Além do casamento de classes-morfossintáticas de palavras, os métodos de mineração de padrões morfo-semânticos propostos por nosso grupo de pesquisas (GOULARTE et al., 2020; SORATO; GOULARTE; FILETO, 2020) usam compatibilidade ou proximidade dos sentidos das palavras, os quais podem ser capturados aplicando reconhecimento de entidades nomeadas (tarefa descrita a seguir) e/ou usando *embeddings* (descritos na seção seguinte) para calcular similaridades ou distâncias semânticas entre palavras.

A figura 2 ilustra algumas entidades identificadas em um trecho de texto clínico usando a ferramenta Babelfy (MORO; RAGANATO; NAVIGLI, 2014), uma ferramenta de domínio aberto para reconhecer entidades e conceitos em textos quaisquer e ligá-los a suas definições no grafo de conhecimento Bebelnet (NAVIGLI et al., 2021), que é uma composição da DBpedia com a WordNet (FELLBAUM, 1998) em vários idiomas. Note que a Babelfy, embora não seja uma ferramenta específica para efetuar NER em textos da área médica, foi capaz de reconhecer medicamentos como "Cetoprofeno", "Triclosan"e "Momentasona", além de ligá-los corretamente a suas descrições na Babelnet. Também conseguiu reconhecer o procedimento "descolonização"e "pele", que pode ser considerada parte da anatomia humana.

Figura 2 – Exemplo de NER em texto clínico usando a ferramenta Babelfy.



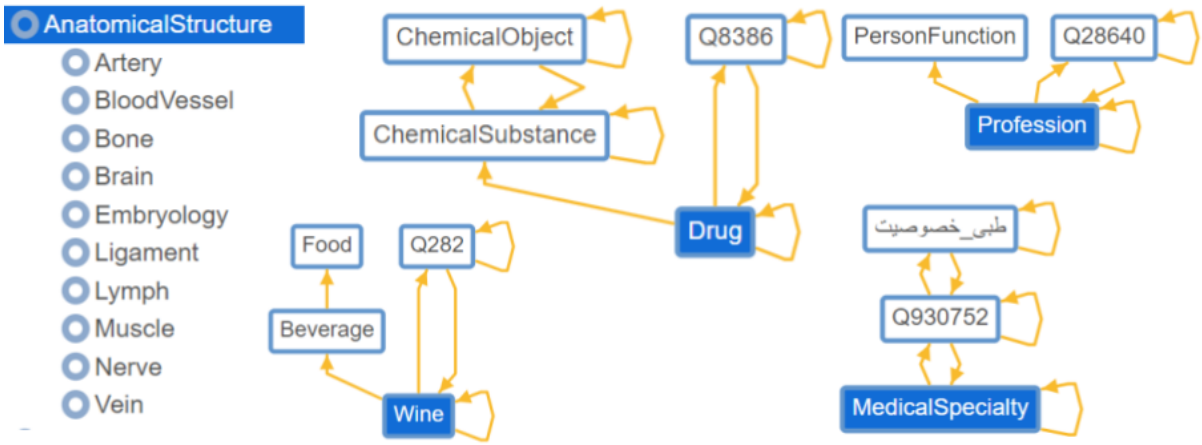
Fonte: o autor.

A figura 3 ilustra algumas classes da Babelnet relacionadas a saúde e alimentação que podem ser consideradas para o reconhecimento de conceitos de conceitos de interesse para mineração de padrões em textos clínicos. Todavia, por serem de cunho geral e produzidas a partir da Wikipedia tais descrições de classes e NER baseado nelas pode não ser suficientemente preciso para aplicações médicas. Na área médica, existem bases de conhecimento específicas, curadas e de alta qualidade tais como a base Descritores em Ciências da Saúde (DeCS), com um pequeno trecho ilustrado na figura 4.

Todavia, ainda há atualmente uma grande carência de ferramentas que explorem tais bases de conhecimento para efetuar NER em textos da área de saúde em língua Portuguesa. Uma das poucas ferramentas específicas para identificar entidades descritas no DeCS em textos da área de saúde é o DeCSfinder[3], também desenvolvido e suportado pelo Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde, também conhecido como BIREME, um centro especializado da Organização Pan-Americana da Saúde / Organização Mundial da Saúde (OPAS/OMS). Resultados da aplicação do DeCSfinder a alguns textos clínicos são ilustrados na figura 5. Note que ele foi capaz de reconhecer, classificar e ligar ao DeCS referências a medicamentos, "pele"e ao termo "Relato de Casos", mas em experimentos preliminares

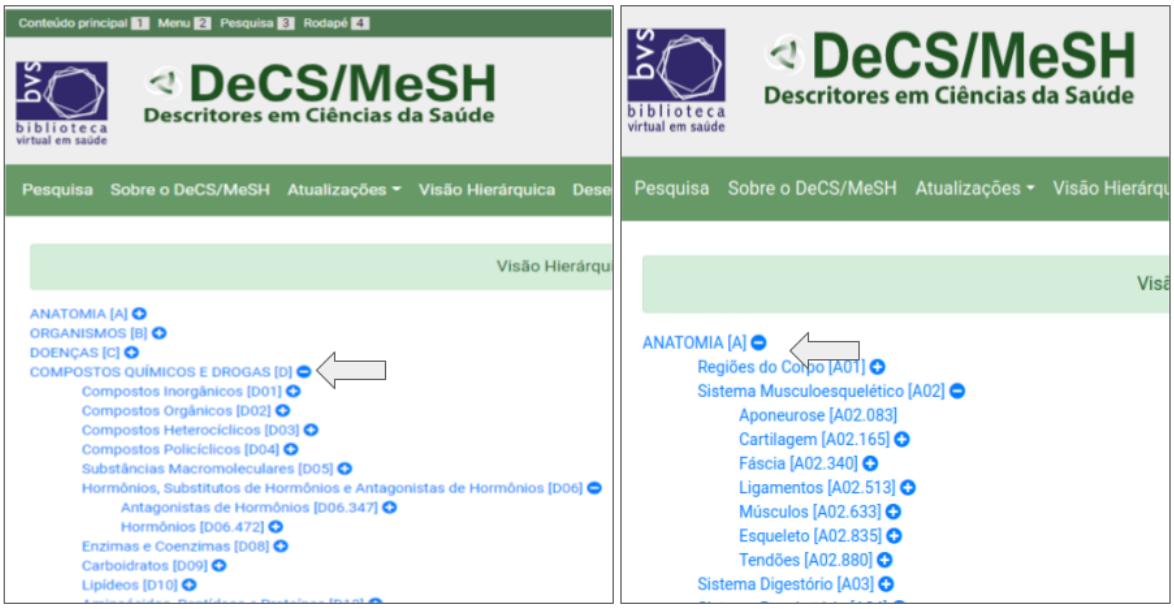
[3] https://decsfinder.bvsalud.org/dmfs

Figura 3 – Exemplos de classes de interesse clínico na Babelnet.



Fonte: o autor.

Figura 4 – Exemplo de classes DeCS.

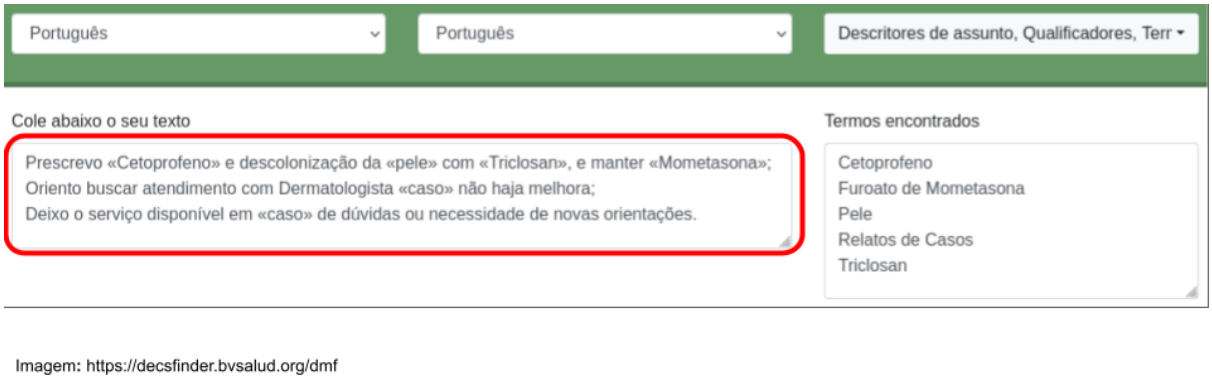


Fonte: o autor.

com nossos textos clínicos tal ferramenta não apresentou robustez, qualidade de resultado (precisão, cobertura, acurácia) e principalmente funcionalidades e conveniência de uso suficientes e adequadas para suprir as nossas necessidades.

Outra possibilidade é usar as atuais soluções para NER baseadas em modelos contextualizados de linguagem treinados com textos da área de saúde e biologia em língua portuguesa, tais como o bioBERTpt, que também teve sintonia fina para efetuar NER em tais tipos de textos. A figura 6 ilustra o resultado de NER com o bioBERTpt em alguns textos fornecidos como exemplo com a própria ferramenta. Note que o bioBERTpt foi capaz de reconhecer entidades

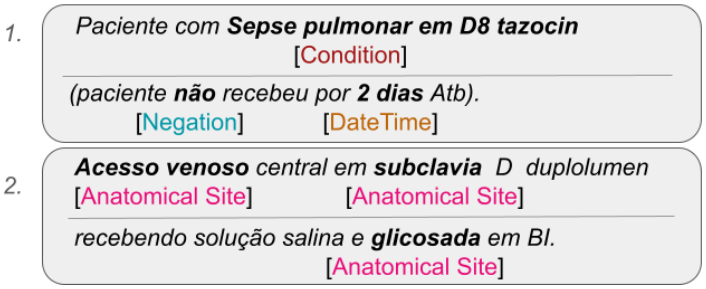
Figura 5 – Exemplo de reconhecimento de entidades com o DeCS finder.



Fonte: o autor.

de interesse na área de saúde com nomes compostos de várias palavras em tais textos.'

Figura 6 – Exemplo de NER em textos clínicos usando o bioBERTpt.



Fonte: o autor.

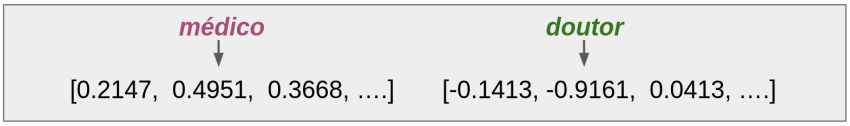
As ferramentas de NER podem ser usadas, por exemplo, na implementação em sistemas de busca e extração de informação. Estes sistemas auxiliam profissionais da área médica a navegar por palavras chave em publicações ao realizar buscas por seus temas de interesse. Segundo Campillos-Llanos et al. (2021), o acesso a tipos específicos de publicações poderia ser mais rápido se os profissionais pudessem customizar sua busca e restringi-la às classes semânticas escolhidas. Em seu artigo, o autor separa as entidades em 4 classes, denominadas Grupos Semânticos, do Sistema Médico Unificado de Linguagem (do inglês, *Unified Medical Language System* UMLS): entidades relativas a patologias (DISO), entidades anatômicas (ANAT), substâncias bioquímicas ou farmacológicas (CHEM) e procedimentos diagnósticos ou terapêuticos e exames laboratoriais (PROC).

2.3 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 7 ilustra *embeddings* de duas palavras distintas, mas cuja semântica pode ser considerada similar, dependendo do contexto em que são usadas: “médico” e “doutor”.

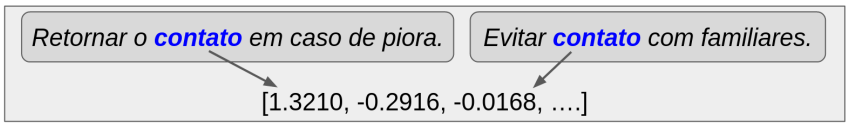
Figura 7 – *Embeddings* das palavras



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 seu significado ou mesmo classe morfossintática mude em diferentes contextos onde é usada, como ilustrado na Figura 8.

Figura 8 – *Embedding* estáticos de uma mesma palavra em contextos diferentes



Fonte: o autor.

Na Figura 8 é possível ver que a palavra **“contato”** tem sentidos distintos nas duas sentenças. Na sentença da esquerda, tal palavra faz o papel de substantivo e tem o sentido de comunicar, enquanto na sentença da direta tem o sentido de toque/proximidade. 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.

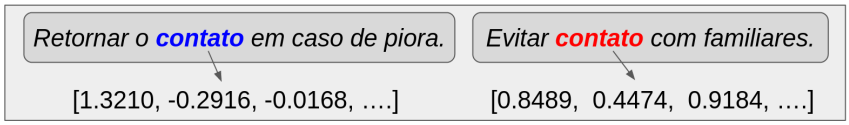
2.3.1 Modelos Contextualizados de Linguagem

Estudos recentes têm buscado meios de automatizar a compreensão da comunicação em linguagem natural através da aplicação de Modelo Contextualizado de Linguagem (MCL) (do inglês, *Contextualized Language Model*) (ZHANG; ZHAO; WANG, 2020). Um MCL como o BERT captura várias características da linguagem, incluindo nuances de significados que variam de acordo com o uso das palavras (DEVLIN et al., 2019), gerando *embeddings* ajustados a cada contexto textual onde ocorre um mesmo léxico. O BERT tem propiciado ganhos consideráveis de desempenho em diversas tarefas de PLN (DEVLIN et al., 2019), tais como segmentação (MULLER; BRAUD; MOREY, 2019) e comparação semântica de senten-

ças (REIMERS; GUREVYCH, 2019) e classificação de documentos (OSTENDORFF et al., 2020).

Atuais MCLs como o BERT (DEVLIN et al., 2019) utilizam vetores derivados de uma rede bidirecional e conseguem capturar o contexto em que uma palavra ocorre em um texto, capturando assim o seu significado específico naquele contexto. Para ilustrar, considere a Figura 9 com duas sentenças que usam a palavra *"contato"*:

Figura 9 – *Embeddings* contextualizados de uma mesma palavra em contextos diferentes



Fonte: o autor.

Na primeira sentença, o contexto onde ocorre a palavra *contato* é delimitado pela palavra "caso", que está a direita, permitindo assim desambiguá-la corretamente, e a um MCL construir uma representação vetorial desta ocorrência de "contato" considerando este contexto. Na segunda sentença o contexto é delimitado pela palavra "Evitar", que está a esquerda. Note que usar somente o contexto da esquerda ou da direita da palavra pode gerar um erro de interpretação. Isso ilustra como a bidirecionalidade de um MCL é essencial para capturar corretamente o significado das palavras em cada contexto (ZHANG; ZHAO; WANG, 2020).

Este trabalho adota BERT como MCL para representar linguagem e gerar *embeddings* para a mineração de padrões, devido à disponibilidade gratuita de seus modelos pré-treinados e por conveniência do seu uso no Google Colaboratory, que disponibiliza acesso direto aos modelos através de bibliotecas específicas.

2.3.2 BERT: *Bidirectional Encoder Representations from Transformers*

Proposto por Devlin et al. (2019), o BERT, acrônimo de *Bidirectional Encoder Representations from Transformers*, é uma rede neural profunda projetada para treinar modelos de linguagem através do processamento bidirecional de documentos não rotulados, considerando os contextos das palavras em ambas as direções. Por se tratar de um modelo flexível, o resultado do pré-treinamento pode sofrer ajustes finos (do inglês, *Fine-Tuning*) com a adição de uma camada de saída, para criação de novos modelos ajustados para realizar tarefas *downstream*, sem a necessidade de modificações significativas na sua arquitetura. Desta forma, o BERT é capaz de lidar com uma variedade de tarefas, incluindo *Question Answering* (QA), classificação e inferências sobre linguagem (DEVLIN et al., 2019).

O BERT adota um pacote padronizado para a entrada das sentenças usadas nas diversas tarefas que pode realizar. Tal pacote converte sentenças de entrada em sequências de tokens, os quais podem se referir a palavras ou sub-palavras. Utilizando o algoritmo *WordPiece*, cada palavra é representada por um token ou por uma sequência de tokens referentes às

suas sub-palavras (DEVLIN et al., 2019). O algoritmo é treinado para minimizar a quantidade de tokens necessária para representar todo o corpo textual, pois várias palavras podem ser construídas pela combinação de sub-palavras, desta forma os *N* tokens mais frequentes vão compor o vocabulário do BERT (WU et al., 2016). Desta forma, o BERT consegue lidar com palavras desconhecidas, i.e. fora do vocabulário (do inglês, *Out-Of-Vocabulary* - OOV), evitando transformar todas elas no mesmo símbolo padrão, o que acarretaria perda de informação.

O BERT original de Devlin et al. (2019) vem pré-treinado com um corpus genérico formado por documentos não rotulados, o qual inclui a Wikipedia (2,5 bilhões de palavras) e o Toronto Book Corpus (800 milhões de palavras), ambos em língua inglesa. Os modelos pré-treinados são disponibilizados em dois tamanhos (a.k.a. variações): $BERT_{Base}$ e $BERT_{Large}$.

A Subseção 2.3.3 apresenta algumas funções de distância e de similaridade entre *embeddings* contextualizados de palavras gerados por um modelo do BERT pré-treinado para o português, o *BERTimbau*. Esses *embeddings* e funções são posteriormente utilizadas em nossa proposta descrita no Capítulo 4 para minerar padrões morfo-semânticos em textos clínicos.

2.3.3 Distâncias e similaridade entre *embeddings*

A mineração de padrões morfo-semânticos que podem estar presentes nos textos clínicos requer determinar elementos (palavras, trechos curtos em torno de palavras) semanticamente próximos uns dos outros. No decorrer desta seção, as métricas (VALLERIAN, 2021) de distância e similaridade utilizadas neste trabalho são apresentadas.

Distância Euclidiana

A Equação 2.1 determina a distância Euclidiana (também conhecida com L2) entre dois vetores. Essa métrica indica o quão próximos dois pontos (*x*,*y*) estão em um plano, tomando o comprimento do caminho mais curto entre eles. Na Figura 10 [4] temos um exemplo de representação da distância Euclidiana entre dois pontos em um plano.

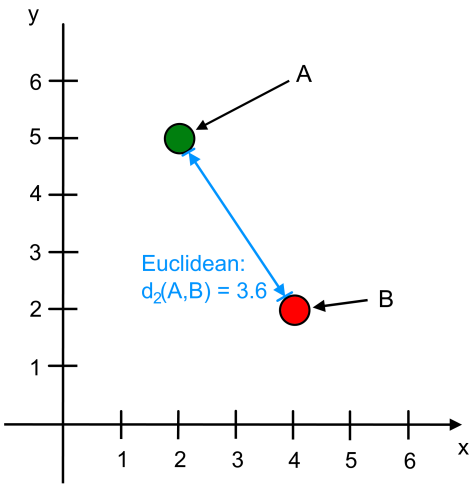
$$dist_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

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 conhecida por *City-Block*. A Figura 11 ilustra a L2 entre dois pontos em um plano.

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

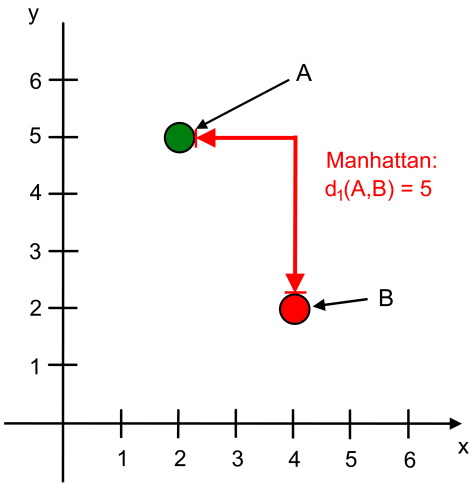
Figura 10 – Distância Euclidiana



Fonte: baseado em What... (2020)

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

Figura 11 – Distância Manhattan



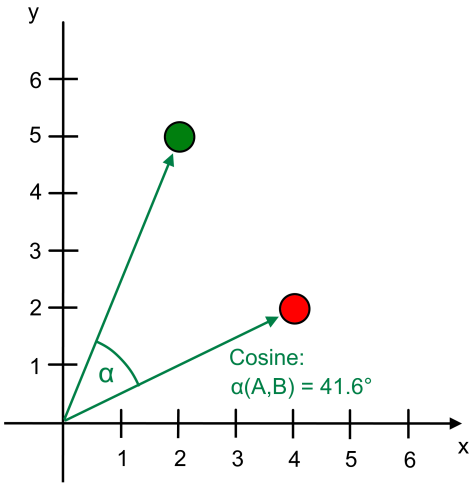
Fonte: baseado em What... (2020)

Similaridade de cosseno

A similaridade de cosseno conforme Equação 2.3 difere das métricas anteriores por não ser uma métrica verdadeira (VALLERIAN, 2021). 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 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 12 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 12 – 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.

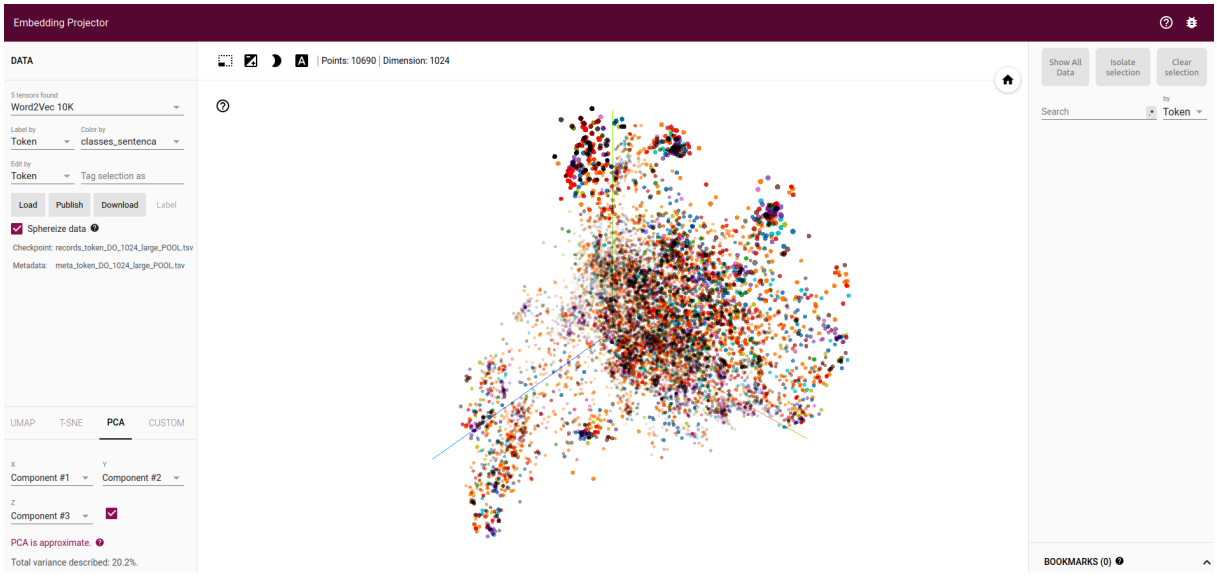
2.4 EMBEDDING PROJECTOR

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 13 ilustra a sua interface gráfica.

2.4.1 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.

Figura 13 – *Embedding Projector*



Fonte: o autor.

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. O método 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 inglês, 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 custumizavel (CUSTOM)

Finalmente, usando projeção custumizavel (*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.

Com o intuito de cunhar uma definição formal, Ehrlinger e Wöß (2016) estabeleceu que "um grafo de conhecimento adquire e integra informações em uma ontologia e aplica um raciocinador para obter novos conhecimentos". Estas informações integradas, fornecem uma maneira melhor de gerenciar e utilizar grandes quantidades de informação, a aplicação de grafos de conhecimento expandiu-se gradualmente para vários campos (XIE et al., 2021). Dentre outros campos, podemos citar: Sistemas de recomendação, Busca Semântica, Resolução de Ambiguidades e Extração de Relação entre entidades. Além disso, "os grafos de conhecimento ganharam recentemente destaque na construção de sistemas de recomendações explicáveis, pois a estrutura do grafo empodera a capacidade de traçar caminhos de raciocínio por trás das recomendações." (XIAN et al., 2020).

Embora um grafo de conhecimento típico pode conter milhões de entidades e bilhões de fatos relacionais, geralmente está longe de ser completo (LIN et al., 2015). Essa limitação afeta a capacidade de prever relações entre entidades e encontrar novos fatos relacionais para as entidades contidas no grafo. Nas seções a seguir vamos exemplificar 2 grafos de conhecimento: um de propósito geral e outro com foco em ontologias médicas.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma breve revisão bibliográfica de trabalhos que realizam extração de informação e mineração de textos com possibilidades de aplicação em classificação e triagem de textos clínicos. A busca inicialmente envolveu publicações a partir de 2020 nos repositórios da *ACM Digital Library*, *IEEE Xplore Digital Library* e no *Google Scholar*. As publicações encontradas foram selecionadas conforme 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 seis publicações com alta relevância para o tema deste trabalho. Elas são discutidas a seguir.

A dissertação de Benício (2020) propôs uma ferramenta para recuperar termos clínicos das anamneses (entrevistas realizadas por médicos com pacientes durante consultas) e estruturá-los para relacionar com os padrões do diagnóstico patológico para posterior utilização em estudos complementares. A ferramenta proposta foi utilizada em dados de anamneses de pacientes de obstetrícia (grávidas e puérperas). No pre-processamento, os dados das anamneses foram submetidos aos processos de tokenização e *stemming*, resultando em uma lista de tokens. Essa lista de tokens é iterada, sendo cada token comparado com uma lista de 12 características desejadas. Caso 1 token correspondente seja encontrado, os tokens seguintes são pesquisados em um dicionário de nomes de superfície obtidos do vocabulário controlado DeCS, o qual pode ser acessado através de uma API. No caso de não haver correspondência direta, a medida de similaridade léxica de *Levenshtein* é utilizada para comparar o token com uma lista de termos previamente construída, com a intenção de reparar um provável erro ortográfico, e o resultado é novamente pesquisado na API do DeCS. Os dados resultantes deste procedimento são guardados para futura análise. Para analisar o desempenho da ferramenta, 4 médicos obstetras, separadamente, realizaram a identificação manual das 12 características desejadas em 30 anamneses e os resultados foram comparados com o resultado obtido, nas mesmas 30 anamneses, pela ferramenta desenvolvida. Após a avaliação manual das anamneses pelos médicos e pelo sistema, foi aplicado o teste estatístico de Kruskal-Wallis, sendo aceita a hipótese de não haver diferenças significativas entre os grupos nos dois testes. Portanto, o trabalho de Benício (2020) utiliza dados clínicos para extrair diagnósticos de textos clínicos, utilizando medidas de similaridade (*Levenshtein*). Porém, não utiliza *embeddings* e nem minera padrões morfo-semânticos, resumindo-se ao reconhecimento de termos relacionados a diagnósticos (como nomes de doenças e condições clínicas), de acordo com a similaridade léxica de suas menções nos textos clínicos com nomes de superfície obtidos do DeCS.

Bertalan e Ruiz (2020) desenvolveu um modelo para predizer resultados judiciais. Para isso, foi construído um corpus de sentenças judiciais, coletadas do sistema eletrônico do Tribunal de Justiça (eSAJ) do Estado de São Paulo. Para restringir a quantidade de documentos, foram selecionados processos classificados como “homicídio simples” e “corrupção ativa”. O

pré-processamento dos textos foi feito utilizando a ferramenta NLTK para realizar o tokenização e remoção de *stop-words*. Em seguida as sentenças judiciais passaram por um processo manual de rotulação (*labeling*), no qual foram classificadas como condenação ou absolvição. Essa classificação foi utilizada para treinar e avaliar modelos supervisionados de ML. Para gerar a representação numérica das palavras foram testadas duas técnicas: medida de frequência do termo–inverso da frequência nos documentos (do inglês, *Term Frequency–Inverse Document Frequency* - TFIDF) e o modelo GloVe pre-treinado para português, gerando os dados para treinamento e teste. Estes dados foram utilizados como entrada para testar 11 classificadores separados em 2 grupos: *Non-Neural Networks* (*Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Classification and Regression Trees, Naïve Bayes, Support Vector Machines*) e *Neural Networks* (*Multilayer Perceptron, Recurrent Neural Networks, Long Short Term Memory, Gated Recurring Unit Networks, Hierarchical Attention Networks*). O trabalho concluiu que os melhores resultados, para o dataset usado, foram obtidos com os classificadores *Regression Trees, Gated Recurring Unit* e *Hierarchical Attention Networks*, de forma que não há uma escolha única entre esses três métodos, pois eles trazem vantagens e falhas particulares.

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.

Santos (2021) avaliou técnicas de ML supervisionado na geração de sugestões de diagnósticos com base na análise dos registros médicos. Devido ao tempo disponível, os autores consideraram apenas doenças do tipo infecciosas e parasitárias. O trabalho analisa os registros médicos e as relações com o DeCS para extrair dados das doenças e da anatomia. Os registros foram pré-processados para gerar uma representação numérica das palavras utilizando as técnicas de *"Label Encoding"* e *"One Hot Encoding"*, da biblioteca *"scikit-learn"*, gerando os dados para treinamento e teste. Estes dados foram utilizados como entrada para os classificadores: *Random Forest*, *Multilayer Perceptron* e *K-neighbors*. O classificador *Multilayer Perceptron* obteve os melhores resultados nos testes ao utilizar um grande *dataset* de treino. Para *datasets* menores, o classificador *K-neighbors* obteve os melhores resultados. O autor sugere utilizar futuramente o ajuste fino do BERT para a classificação. Apesar de não classificar padrões morfossintáticos, o trabalho fornece uma representação dos registros médicos e realiza a sua classificação para sugestões de diagnósticos.

Martins (2021) propôs um protótipo para extrair e calcular indicadores financeiros de uma empresa a partir dos seus relatórios. Esses relatórios estão dispostos em arquivos PDF, obtidos manualmente dos sites das empresas escolhidas. Para extrair os textos dos arquivos PDF, duas técnicas foram avaliadas: uma biblioteca de leitura de arquivos PDF, chamada *PyPDF2*, e a conversão dos arquivos PDF para imagens para uso de reconhecimento ótico de caracteres (do inglês, *Optical Character Recognition* - OCR). Para o pré-processando do texto, foi utilizado a ferramenta NLTK, realizando a tokenização e o *stemming* das palavras. O resultado deste pre-processamento é armazenado em uma estrutura de dados onde são pesquisados os indicadores financeiros lucro líquido e patrimonio líquido. Essa pesquisa é feita iterando pela estrutura de dados, buscando por tokens numéricos e, ao encontrar, verificando se os tokens anteriores estão em uma lista de possíveis representações do indicador financeiro desejado. Caso os 2 indicadores sejam encontrados em um relatório, é possível calcular um terceiro indicador: retorno sobre o patrimônio líquido (do inglês, *return on equity* - ROE). Para avaliar o protótipo foram realizados 5 experimentos, alterando os parâmetros das bibliotecas de leitura, e o melhor resultado foi obtido usando a técnica de OCR para extração do texto.

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 2 fornece um resumo comparativo das propostas selecionadas na literatura. Os trabalhos são ordenados cronologicamente nas linhas da tabela. Eles são comparados conforme os aspectos das soluções de PLN que consideramos mais relevantes, o quais aparecem nas colunas. A primeira coluna **Autor e Ano** define o nome dos autores e ano de publicação do trabalho avaliado. A segunda coluna **Área de aplicação** descreve o domínio dos documentos analisados. A terceira coluna **Objetivos** refere-se à proposta de cada trabalho. O tipo de ***Embedding*** utilizado por cada trabalho é indicado na quarta coluna. A quinta coluna **Abordagem** refere-se aos métodos utilizados na proposta. Por fim, a coluna **Ferramentas** cita as ferramentas utilizadas no desenvolvimento da proposta.

Tabela 2 – Tabela comparativa

Autor e Ano	**Área de aplicação**	**Objetivos**	***Embeddings***	**Abordagens**	**Ferramentas**
Benício (2020)	Médica / Obstetrícia	Detectar termos clínicos	N/A	*Stemming*, *Levenshtein*	DeCS
Bertalan e Ruiz (2020)	Juridica	Predizer resultados judiciais	GloVe	*Labeling*, TF-IDF, ML supervisionado	NLTK
Sorato, Goularte e Fileto (2020)	Posts em Micro-blogs	Minerar padrões semânticos para analisar e classificar discursos	GloVe	*PoS-Tagging*, TF-IDF	spaCy, Scikitlearn, NLTK
Santos (2021)	Médica	Sugestão de diagnóstico	N/A	*Label Encoding*, *One Hot Encoding*, ML supervisionado	NLTK, spaCy, DeCS
Martins (2021)	Financeira	Detectar identificadores financeiros	N/A	OCR, *Stemming*	NLTK
Braz-Junior e Fileto (2021)	Perguntas de QA	Analisar coerência de discursos	BERT	*POS-Tagging*	*nlpnet*, *scipy*, Google Colab
Nosso trabalho	**Médica**	**Minerar Padrões morfo-semânticos**	**BERT**	**ML não supervisionado**	**spaCy, DBpedia, Google Colab**

Fonte: o autor.

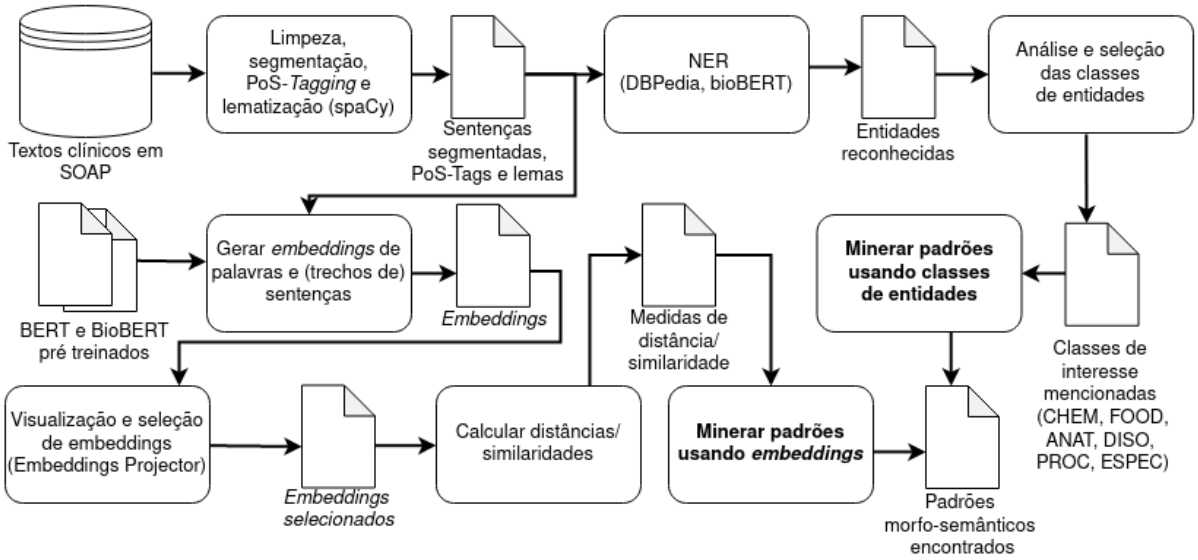
Nenhum dos trabalhos analisados usa *embeddings* contextualizados. Bertalan e Ruiz (2020) e Sorato, Goularte e Fileto (2020) utilizam *embeddings* estáticos como GloVe para,

respectivamente, predizer resultados judiciais e minerar padrões em textos, enquanto Benício (2020), Santos (2021) e Martins (2021) não usam *embedding* nenhum. Nossa proposta explora NER para identificar e classificar certas menções relevantes em textos (a medicamentos, comidas, estruturas anatômicas, especialidades e procedimentos médicos). Alternativamente, usa *embeddings* contextualizados produzidos pelo BERT (que capturam contexto textual em ambas as direções). As classes de NER e os *embeddings* permitem determinar casamento ou similaridade de sentidos, respectivamente, em métodos alternativos para minerar padrões morfo-semânticos em textos.

4 PROCESSO E ALGORITMOS PARA MINERAR PADRÕES

Este capítulo descreve as etapas do processo e algoritmos propostos neste trabalho para minerar padrões morfo-semânticos em textos clínicos. A Figura 14 ilustra o fluxo de informação proposto e avaliado neste trabalho para minerar os padrões em textos clínicos no padrão SOAP cedidos por uma empresa que presta serviços a operadoras de plano de saúde, embora o processo proposto possa ser aplicado a textos de outras fontes.

Figura 14 – Processo para mineração de padrões morfo-semânticos em textos clínicos.



Fonte: o autor.

O processo se inicia com o uso do spaCy para efetuar segmentação do texto em sentenças, POS-*Tagging* e lematização. O resultado desta primeira tarefa é utilizado em 2 tarefas subsequentes em paralelo. Na primeira tarefa, NER, as sentenças de todos os documentos são submetidas a duas soluções de NER (consultas SPARQL à DBpedia e bioBERT NER). Os resultados obtidos são comparados e as entidades reconhecidas são agrupadas de acordo com classes de interesse para minerar padrões: medicamentos (CHEM), comidas (FOOD), estruturas anatômicas (ANAT), doenças (DISO), procedimentos médicos (PROC) e especialidades médicas (ESPEC). Na segunda tarefa que usa sentenças segmentadas, o *BERTimbau* é usado para gerar embeddings das palavras de todos os documentos 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 clínicos desenvolvidos no âmbito deste trabalho. Tais algoritmos avaliam a similaridade semântica das palavras usadas nos documentos em torno das entidades mencionadas pertencentes às classes de interesse para identificar os padrões semânticos, ou seja, instâncias de texto com os mesmos sentidos em torno das mesmas entidades mencionadas, embora muitas vezes usando construções léxicas e

sintáticas distintas. Uma forma alternativa de minerar padrões morfo-semânticos é de acordo com correspondências de classes de sentidos obtidas através de NER. As tarefas do processo proposto são descritas em mais detalhes a seguir.

4.1 LIMPEZA, SEGMENTAÇÃO DE SENTENÇAS, *POS-TAGGING* E LEMATIZAÇÃO

A primeira etapa do processo proposto é o pre-processamento dos documentos (Textos Clínicos). Nesta etapa primeiro se faz a padronização dos textos, mediante remoção de certos caracteres especiais (e.g.,$\backslash n$), 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 documento, lista de seus tokens, lista de lemmas, lista de POS-*Tags*, lista de verbos e a lista de sentenças do documento. Esses dados são utilizadas para facilitar processo de geração dos *embedding* e reconhecimento de entidades nomeadas relevantes para mineração de padrões.

4.2 RECONHECIMENTO DE ENTIDADES - NER

A tarefa de NER é realizada utilizando os dados gerados na etapa anterior (descrita na Seção 4.1). As sentenças dos documentos são submetidas a duas soluções de NER para a identificação das entidades clínicas. A primeira é o *BioBERTpt*, o qual consiste de um modelo do BERT com ajuste fino para efetuar NER em textos do domínio clínico na língua portuguesa. O uso desta ferramenta de NER foi feito seguindo o exemplo postado pelo autor em seu repositório no github [1]. A segunda solução utilizada foi consultas pelas palavras encontradas no textos clínicos na DBPedia através de seu *endpoint* SPARQL.

A Figura 15 apresenta um exemplo de consulta desenvolvida em nosso trabalho buscando recursos rotulados pela palavra "Cetoprofeno". Como era desejado que fossem reconhecidos apenas recursos de ontologias registradas em português, criamos uma consulta que filtra os resultados retornados que contenham estas características. O filtro para ontologias verifica se o tipo retornado pela DBpedia contem a *string* "*http://dbpedia.org/ontology/*". Já o parâmetro "*@pt*" verifica se a palavra passada na consulta tem correspondência em português. Para haver correspondência também é preciso que a palavra esteja capitalizada, ou seja, com a primeira letra maiúscula e as demais minúsculas. Ainda assim, é possível que a palavra corresponda com múltiplas entidades de áreas diferentes, retornando uma lista extensa de resultados de diferentes áreas de conhecimento.

Estas consultas retornam todos os tipos de recursos da DBpedia cujos rótulos são as respectivas palavras dos textos clínicos. Após o término desses procedimentos, os dados ob-

[1] Disponível em https://github.com/HAILab-PUCPR/BioBERTpt

Figura 15 – Consulta SPARQL feita na DBpedia para a palavra 'Cetoprofeno'

```
select distinct ?type
where {
     ?i rdfs:label "Cetoprofeno"@pt ;
     a ?type .
     FILTER (strstarts(str(?type),str("http://dbpedia.org/ontology/")))
     FILTER (strstarts(str(?i), str("http://dbpedia.org/resource/Ketoprofen")))
}
```

Fonte: o autor.

tidos são armazenados em uma estrutura contendo para cada palavra consultada o resultado obtido pelo NER. Esses resultados são utilizados para realizar a classificação das entidades reconhecidas.

4.3 ANÁLISE E SELEÇÃO DAS CLASSES DE ENTIDADES

Nesta etapa, verificamos os resultados obtidos pelas 2 ferramentas de NER utilizadas na Seção 4.2. O objetivo desta verificação é selecionar a ferramenta que reconhece corretamente a maior quantidade de entidades mencionadas nos textos clínicos. Após a seleção da ferramenta, as entidades reconhecidas por ela são separadas em classes de interesse, através de uma função desenvolvida em nosso código, utilizando a classificação proposta por Campillos-Llanos et al. (2021). Em nosso trabalho foram adicionados outras 2 classes de interesse: especialidades médicas (SPEC) e alimentos (FOOD). Estas 2 classes foram adicionadas pois verificamos que em nossos documentos os profissionais realizam orientações de ingestão de alimentos e indicações para outras especialidades médicas. Logo, poderíamos minerar padrões usando essas 2 classes de interesse adicionadas. Após a classificação das entidades nomeadas mencionadas, a lista com a respectiva classe de cada palavra dos documentos é concatenada na estrutura semi-estruturada criada na Seção 4.1. Esta classificação é utilizada no processo de mineração de padrões usando classes de interesse.

4.4 GERAR *EMBEDDINGS* DE PALAVRAS E (TRECHOS DE) SENTENÇAS

Nesta etapa, as sentenças dos documentos 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 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 concatenação dos *embeddings* de seus *tokens*. Janelas são fragmentos de sentenças centrados em uma classe de interesse, 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 tomados como a média (*MEAN pooling*) das 4 últimas camadas do BERT, gerando embeddings com 4.096 valores (dimensões).

Algoritmo 1: Criar lista de janelas

Data: Lista de sentenças("*idSentenca*","*tokens*", "*postagging*", "*entidadeReconhecida*", "*classeDeInteresse*", "*posER*") 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.      posER ← reg["posER"]
6.      entidadeReconhecida ← reg["entidadeReconhecida"]
7.      classeDeInteresse ← reg["classeDeInteresse"]
8.      texto ← juncao(tokens)
9.      embSentenca ← getEmbeddingsText(texto)
10.     tamJanela ← 1
11.     janelaInf, janelaSup, expande ← expandeJanela(token, posER, posER)
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, entidadeReconhecida, classeDeInteresse]
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 classe de interesse. As sentenças relevantes (*SR*) utilizadas nessa tarefa são todas as que possuam uma palavra classificada como descrito na Seção 4.3. Uma lista de registros é criada para armazenar

as janelas com seus *embeddings*. Todas as sentenças são percorridas para gerar os *embeddings*. O registro de cada sentença da lista é composto pelo id da sentença, lista de *tokens*, posição da entidade reconhecida, a própria palavra reconhecida e classe de interesse da entidade reconhecida. 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 e a classe de interesse 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.

4.5 VISUALIZAÇÃO E SELEÇÃO DE *EMBEDDINGS*

Nesta etapa, os dados gerados como explicado na seção 4.4 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., classe morfossintática ou classe de sentido de *embedding* de palavra, classe de interesse de um *embedding* de palavra, classes de interesse presentes na sentença). 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. A Figura 16 apresenta uma lista de características dos *embeddings* dos tokens dos documentos 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.

Figura 16 – Label de característica do *Embeddings Projector*

Fonte: o autor.

O *Embedding Projector* possibilita validar os *embeddings* carregados. Isso pode ser feito usando uma das técnicas apresentadas na Subseção 2.4.1 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. Se o conjunto de dados apresentar alguma propriedade relevante ou desejável, o conjunto de dados validado é separado manualmente para ser utilizado na tarefa na mineração de padrões.

4.6 CALCULAR DISTÂNCIAS/SIMILARIDADES

O cálculo das medidas de similaridade e distância é realizado para cada documento. São calculadas a distância (Euclidiana e Manhathan) e a similaridade (cosseno) para todos os pares de *embeddings* de palavras. Todas as distâncias entre pares de *embeddings* ficam pré-calculadas e armazenadas para reuso pela próxima e última etapa do processo proposta, 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 entidade reconhecida pertencente a uma classe de interesse. Como entrada é utilizada a lista criada no Algoritmo 1 contendo os *embeddings* de janelas formado por id da sentença ("idSentenca"), sentença, janela, tamanho da janela (”*compJanela*”), lista de *embeddings* (”*embJanela*”) e classe de interesse. Todos os elementos da lista de *embeddings* de janelas são percorridos e calculadas as medidas de distâncias/similaridade dos *embeddings* da janelas entre si ($n^2 - n$ comparações). 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 ja-

nela 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, classe de interesse, id da sentença comparada, sentença comparada, tamanho da janela comparada, entidade reconhecida 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*", "*entidadeReconhecida*", "*classeDeInteresse*") de sentenças relevantes (*SR*)

Result: Lista de registro

1. $listaRegistro \leftarrow [\,]$
2. **foreach** $index_a, reg_a$ *in SR* **do**
3. $\quad idSentenca_a \leftarrow reg_a[\text{"}idSentenca\text{"}]$
4. $\quad sentenca_a \leftarrow reg_a[\text{"}sentenca\text{"}]$
5. $\quad janela_a \leftarrow reg_a[\text{"}janela\text{"}]$
6. $\quad compJanela_a \leftarrow reg_a[\text{"}compJanela\text{"}]$
7. $\quad embJanela_a \leftarrow reg_a[\text{"}embJanela\text{"}]$
8. $\quad entidadeReconhecida_a \leftarrow reg_a[\text{"}entidadeReconhecida\text{"}]$
9. $\quad classeDeInteresse_a \leftarrow reg_a[\text{"}classeDeInteresse\text{"}]$
10. $\quad$ **foreach** $index_b, reg_b$ *in SR* **do**
11. $\quad\quad idSentenca_b \leftarrow reg_b[\text{"}idSentenca\text{"}]$
12. $\quad\quad sentenca_b \leftarrow reg_b[\text{"}sentenca\text{"}]$
13. $\quad\quad janela_b \leftarrow reg_b[\text{"}janela\text{"}]$
14. $\quad\quad compJanela_b \leftarrow reg_b[\text{"}compJanela\text{"}]$
15. $\quad\quad embJanela_b \leftarrow reg_b[\text{"}embJanela\text{"}]$
16. $\quad\quad entidadeReconhecida_b \leftarrow reg_b[\text{"}entidadeReconhecida\text{"}]$
17. $\quad\quad classeDeInteresse_b \leftarrow reg_b[\text{"}classeDeInteresse\text{"}]$
18. $\quad\quad$ **if** $idSentenca_a \neq idSentenca_b$ *OR* $janela_a \neq janela_b$ **then**
19. $\quad\quad\quad medida \leftarrow getMeasurementsEmbedding(embJanela_a, embJanela_b)$
20. $\quad\quad\quad regcomp_a \leftarrow [idSentenca_a, sentenca_a, janela_a, compJanela_a,$
$\quad\quad\quad\quad entidadeReconhecida_a, classeDeInteresse_a,$
$\quad\quad\quad\quad idSentenca_b, sentenca_b, janela_b, compJanela_b,$
$\quad\quad\quad\quad entidadeReconhecida_b, classeDeInteresse_b, medida]$
21. $\quad\quad\quad regcomp_b \leftarrow [idSentenca_b, sentenca_b, janela_b, compJanela_b,$
$\quad\quad\quad\quad entidadeReconhecida_b, classeDeInteresse_b,$
$\quad\quad\quad\quad idSentenca_a, sentenca_a, janela_a, compJanela_a,$
$\quad\quad\quad\quad entidadeReconhecida_a, classeDeInteresse_a, medida]$
22. $\quad\quad\quad listaRegistro.append(regcomp_a)$
23. $\quad\quad\quad listaRegistro.append(regcomp_b)$

4.7 MINERAÇÃO DE PADRÕES MORFO-SEMÂNTICOS

Finalmente, a tarefa de minerar padrões morfo-semânticos pode ser realizada em torno de *embeddings* de palavras ou *embeddings* de entidades reconhecidas pertencentes as classes de interesse nas sentenças dos textos clínicos. A tarefa 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 4.6). O Algoritmo 3 apresenta na forma de pseudocódigo o programa criado para a tarefa de mineração de padrões usando medidas de distâncias/similaridade de *embeddings* de palavras de sentenças que contenham uma determinada classe de interesse.

Algoritmo 3: Mineração de Padrões

Data: Lista de medidas de entre janelas("$idSentenca_1$", "$sentenca_1$", "$janela_1$", "$compJanela_1$", "$entidadeReconhecida_1$", "$classeDeInteresse_1$", "$idSentenca_2$", "$sentenca_2$", "$janela_2$", "$compJanela_2$", "$entidadeReconhecida_2$", "$classeDeInteresse_2$", "$medida$")

Result: dicionário de ocorrências de padrões semânticos em janelas

```
1.  dicionario ← dict()
2.  foreach index, reg in listaJanelas do
3.      idSentenca ← reg["idSentenca_1"]
4.      entidadeReconhecida ← reg["entidadeReconhecida_1"]
5.      janela ← reg["janela_1"]
6.      medida ← reg["medida"]
7.      if medida ≥ threshold then
8.          identificador =
              idSentenca + "__" + entidadeReconhecida + "__" + 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
```

O Algoritmo 3 tem como entrada a lista criada no Algoritmo 2 contendo pares de janelas ($janela_1, janela_2$) comparadas e sua medida. Um dicionário vazio é criado para armazenar as ocorrências de janelas e todas as medidas entre janelas são percorridas. Para medida entre janelas delimitado pelo valor do *threshold* é gerado um identificador formado pela concatenação do id da sentença, a classe de interesse e a janela. Este identificador é a chave para a entrada

no dicionário de ocorrências. Se o identificador não existir no dicionário, uma entrada para ele é criada com um conjunto vazio. Para cada ocorrência de um identificador é adicionado o par de janela avaliado ao conjunto de seu respectivo identificador. Por fim a contagem de pares de janelas de cada item do dicionário é realizada para identificar qual janela apresenta a maior ocorrência.

5 EXPERIMENTOS E RESULTADOS

Esta seção relata os experimentos realizados para avaliar a nossa proposta de mineração de padrões morfo-semânticos. Esses experimentos visam verificar a existência e as características de padrões morfo-semânticos nos textos clínicos escritos por profissionais da área da saúde ao realizar atendimentos. Primeiramente, a Seção 5.1 apresenta alguns detalhes da implementação do processo proposto para minerar padrões. Em seguida, a Seção 5.2 descreve o conjunto de dados usado nos experimentos. A Seção 5.3 apresenta os resultados dos experimentos referentes às tarefas iniciais do processo proposto no Capítulo 4 e os resultados dos cálculos das medidas de distância e similaridade. A Seção 5.4 apresenta e comenta os resultados obtidos pelas ferramentas de NER. Finalmente, a Seção 5.5 apresenta visualizações de *embeddings* e os resultados da mineração de padrões.

5.1 IMPLEMENTAÇÃO

A proposta foi implementada 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. Utilizamos dois modelos BERT pré-treinados para a língua portuguesa. Primeiro, o *BERTimbau*[3] (SOUZA; NOGUEIRA; LOTUFO, 2020) no tamanho *large*, no formato "*cased*" (com caracteres maiúsculos e minúsculos), disponível gratuitamente. O segundo modelo foi o *BioBERTpt*[4] (SCHNEIDER et al., 2020) na sua variação *all*, também disponível gratuitamente. Para a tarefa de segmentação de sentenças e POS-*Tagging* das palavras utilizamos a ferramenta de PLN spaCy[5] versão 3.2.0. Os *embeddings* do BERT foram manipulados usando os métodos da biblioteca PyTorch versão 1.8.1.

5.2 CONJUNTO DE DADOS

Os dados de atendimentos médicos disponibilizadas para realização deste trabalho foram fornecidos por uma empresa que presta serviços a operadoras de plano de saúde, com a

[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://github.com/HAILab-PUCPR/BioBERTpt
[5] https://spacy.io/

supervisão de um dos sócios de tal empresa. Os dados foram recebidos em arquivos no formato CSV, contendo apenas os campos de texto livre dos prontuários. Desta forma nomes, documentos e outras informações sensíveis dos pacientes não foram compartilhadas.

Conforme sugerido pelo sócio da empresa ao disponibilizar os dados, resolvemos focar os testes nos 2 campos das notas de evolução do modelo SOAP com informações mais relevantes: Objetivo (O) e Plano (P). Após realizar a extração e a limpeza dos dados, identificamos que o campo Plano (P) possuía textos mais estruturados e em maior quantidade. Desta forma escolhemos usar apenas este campo para a realização do trabalho. Ao extrair o campo Plano do arquivo CSV, foram obtidos 458 documentos.

Os textos livres dos documentos selecionados continham algumas características que precisaram ser ajustadas antes de submetê-los aos modelos de PLN. Foi necessário separar números de unidades de medida ('100mg' para '100 mg') para gerar o reconhecimento correto pelo BERT. Também foi observado que existiam muitos documentos idênticos repetidos nos textos extraídos do campo Plano (89 no total), mostrando que os profissionais reaproveitam os textos em casos de diagnósticos semelhantes. Como documentos idênticos não trazem relevância para a mineração de padrões e também geram uso desnecessário de disco e tempo de processamento, estes documentos foram removidos dos textos clínicos. Após a remoção dos documentos idênticos, a quantidade de documentos ficou em 369. A Tabela 3 apresenta as principais estatísticas encontradas no campo Plano dos textos clínicos depois da remoção dos documentos idênticos.

Tabela 3 – Estatísticas do conjunto de dados

	Sentenças	**Palavras**	**Tokens BERT**	**Palavras desconhecidas**
Quantidade Total	1.054	12.912	19.710	3.722
Média por Doc.	3,81	46,61	71,16	13,44
Desvio padrão por Doc.	2,15	36,89	49,73	8,22

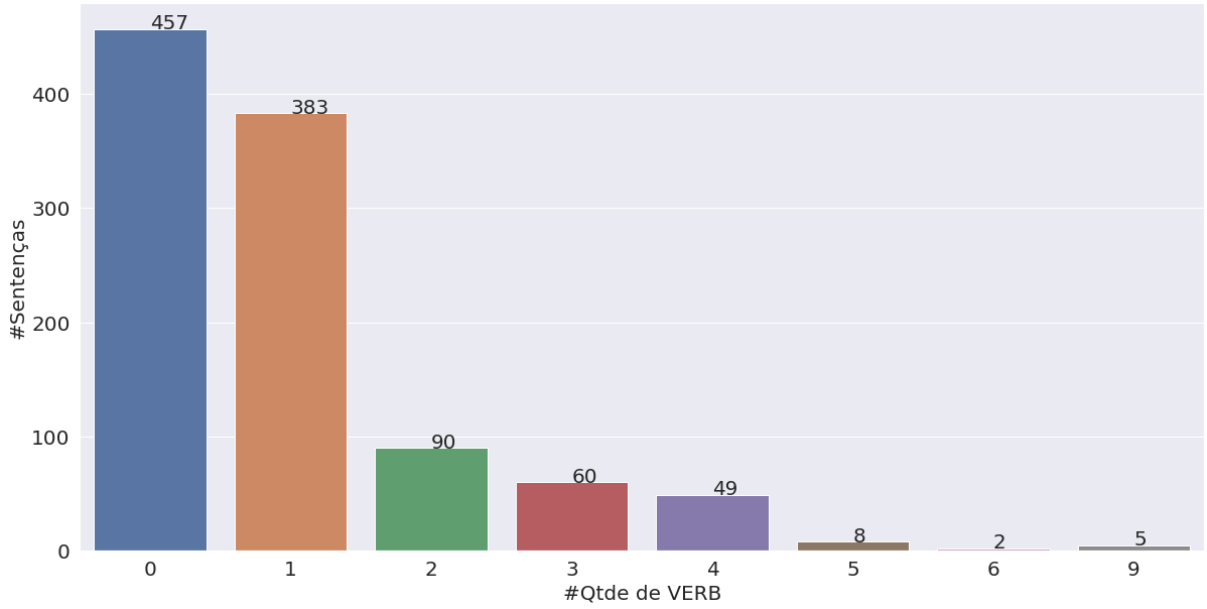
Fonte: o autor.

Com o intuito de tentar sanar a baixa quantidade de documentos e o auto índice de reuso de texto por parte dos profissionais, procuramos novamente o sócio da empresa responsável por disponibilizar os textos clínicos, com o intuito de solicitar uma quantidade maior de documentos. Infelizmente obtivemos uma resposta negativa, pois o profissional havia se desligado da empresa que era associado. Desta forma, os experimentos realizados foram afetados negativamente pela falta de dados e pelo vício textual dos profissionais. Nas seções a seguir apresentamos os resultados das principais tarefas realizadas nos experimentos, ordenadas de acordo com o processo descrito no Capítulo 4.

5.3 POS-TAGGING E MEDIDAS DE DISTÂNCIAS E SIMILARIDADE

Os textos clínicos foram primeiramente processados com o spaCy para efetuar uma análise dos seus conteúdos, como por exemplo a distribuição das classes morfossintáticas em todo o corpus, por documento e por sentença. A Figura 17 ilustra a quantidade de verbos por sentença no campo Plano dos textos clínicos. Podemos observar que a maioria das sentenças não tem verbos e nas sentenças com verbos a maioria tem apenas 1.

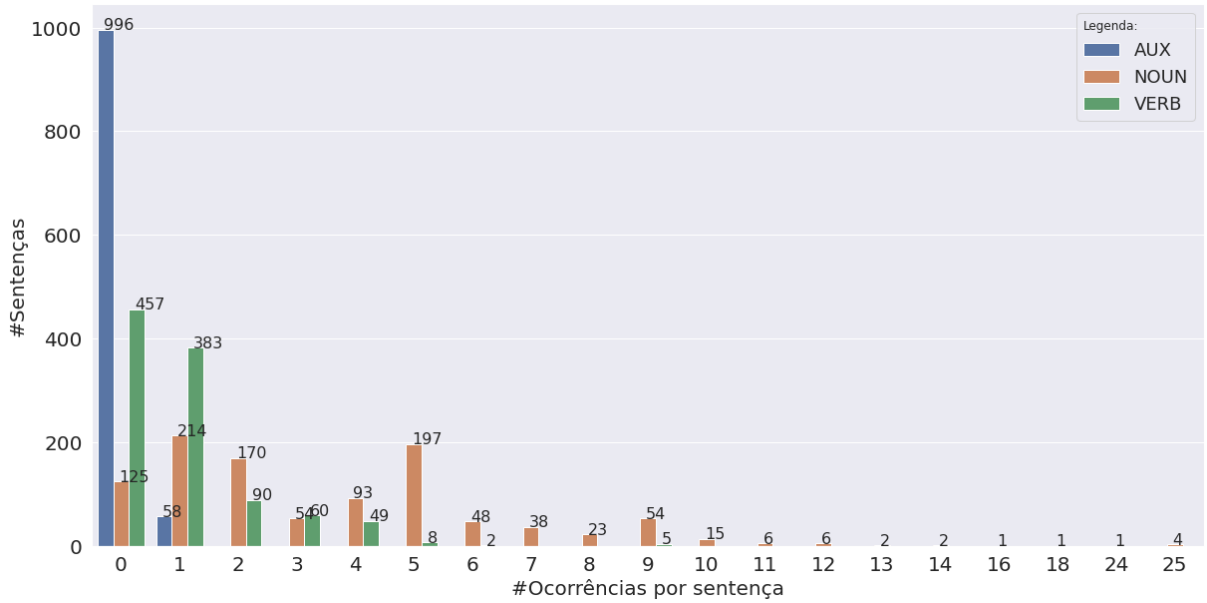
Figura 17 – Quantidade de verbos por sentença



Fonte: o autor.

A Figura 18 apresenta a quantidade das classes morfossintáticas verbo(“VERB”), substantivo(“NOUN”) verbo auxiliar(“AUX”) por sentença nos textos clínicos. Podemos observar que a maioria das sentenças não tem verbos nem verbos auxiliares e a maioria das sentenças tem 1, 2 ou 5 substantivos (214, 170 e 197 respectivamente). Essa quantidade alta de sentenças sem verbos e com muitos substantivos deve-se á característica textual dos profissionais da saúde de mencionar mais de 1 remédio ou procedimento em uma mesma sentença.

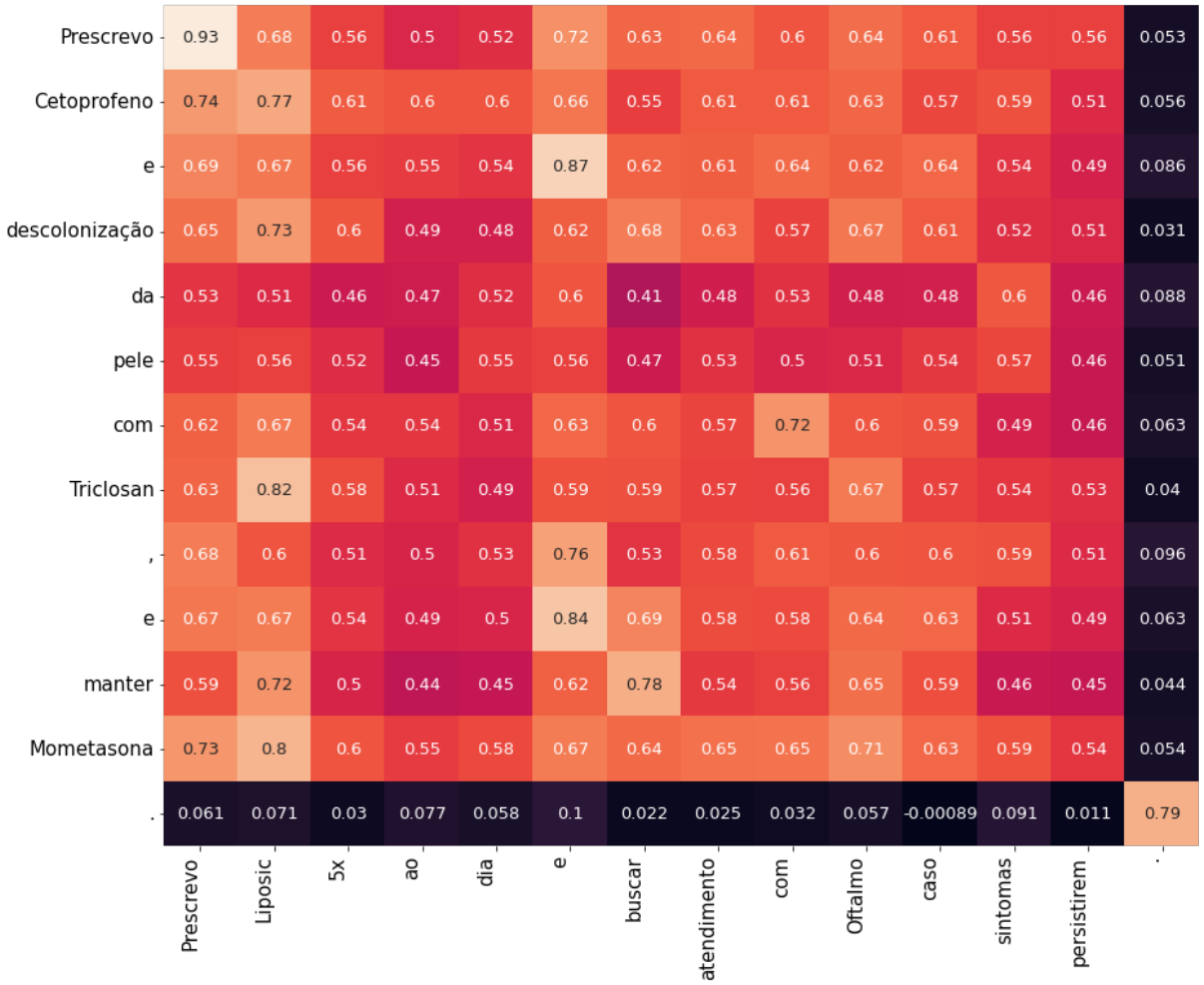
Figura 18 – Quantidade de verbos, verbos auxiliares e substantivo por sentença



Fonte: o autor.

Com a análise estatística preliminar realizada, deu-se início no processo de identificar as relações de similaridade semântica entre os *embeddings* das palavras dos documentos obtidos do campo **PLANO** dos textos clínicos. Durante os experimentos envolvendo *embeddings* contextualizados das palavras, foram testados diferentes gráficos para melhor visualizar a similaridade e distância entre os *embeddings* gerados. Um exemplo de gráfico gerado para este fim é o mapa de calor(do inglês, *Heatmap*) na Figura 19, que ilustra a similaridade do cosseno entre os *embeddings* de duas sentenças distintas de nosso conjunto de dados. Neste mapa, cada célula do mapa possui uma graduação em cores indicando o calor, ou seja, quanto mais diferente (próximo de 0) forem as palavras mais frio (escuro) e quanto mais similar (próximo de 1) mais quente (claro).

Figura 19 – Mapa de calor das medidas de similaridade do cosseno entre *embeddings* das palavras de sentenças distintas.

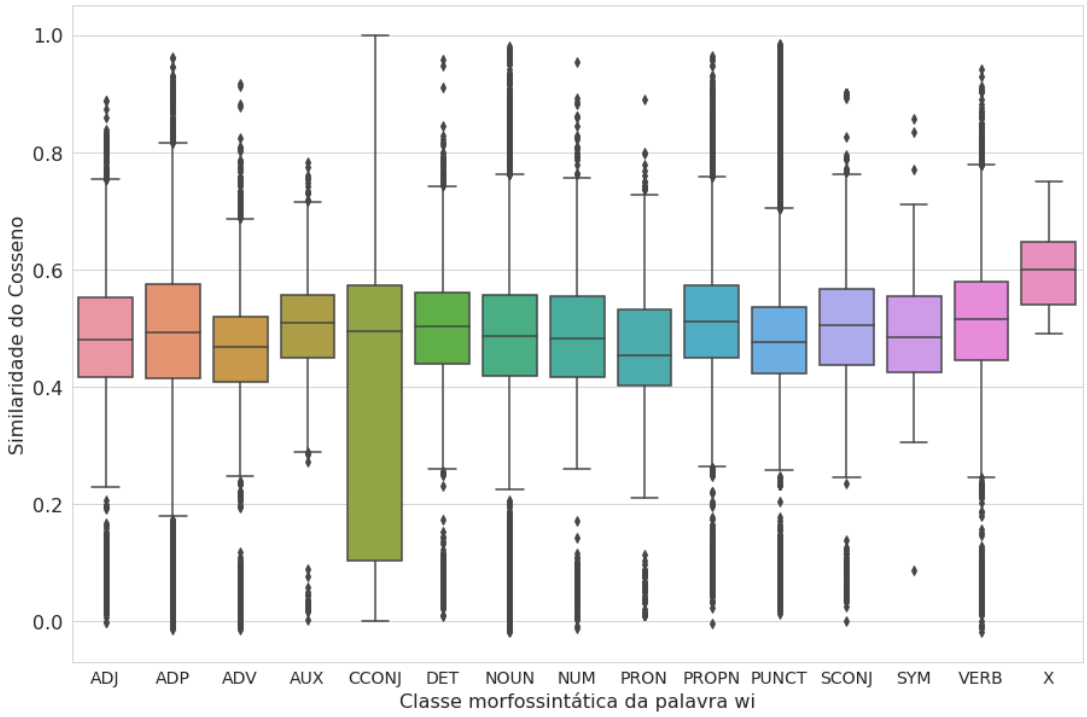


Fonte: o autor.

Na Figura 19 é possível identificar que a menção ao medicamento “Lipsosic” (eixo x) tem uma similaridade igual a $0,82$ com Triclosan (eixo y), $0,8$ com “Mometasona” e $0,77$ com Cetoprofeno. Esta análise inicial fornece indícios que existe certa similaridade entre palavras referentes a medicamentos de duas sentenças.

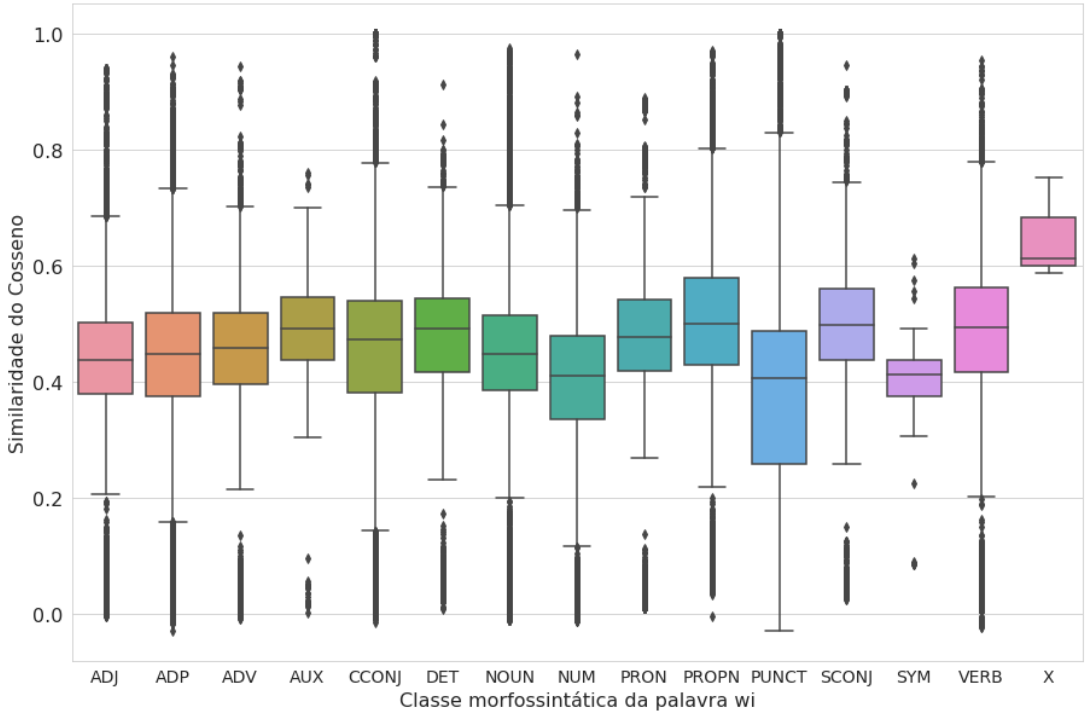
Também realizamos análises preliminares para comparar os resultados das medidas dos *embeddings* gerados por um modelo genérico do BERT (*BERTimbau*) com as mesmas medidas dos *embeddings* gerados por um modelo *fine-tuned* para a área médica (*BioBERT pt*). Em uma destas análises usamos os dois modelos para obter a discrepância das palavras do campo Plano de todos os textos clínicos, separadas por classe morfossintática, utilizando similaridade do cosseno. As figuras 20 e 21 demonstram gráficos de box-plot com os resultados obtidos com o *BERTimbau* e o *BioBERT pt*, respectivamente.

Figura 20 – Distribuição das discrepâncias entre pares (w_i/w_j) de palavras utilizando similaridade do cosseno e agrupada pela classe morfossintática da palavra w_i utilizando o *BERTimbau*.



Fonte: o autor.

Figura 21 – Distribuição das discrepâncias entre pares (w_i/w_j) de palavras utilizando similaridade do cosseno e agrupada pela classe morfossintática da palavra w_i utilizando o *BioBERT pt*.



Fonte: o autor.

A Tabela 4 apresenta os valores estatísticos das comparações feitas entre os *embed-*

dings das palavras dos textos clínicos. Observando essas estatísticas, podemos verificar que até 75% (terceiro quartil- Q3) a similaridade dos *embeddings* fica abaixo de $0,5595$. Com isso deduzimos que existe pouca similaridade entre os *embeddings* contextualizados gerados para as palavras. Logo, os *embeddings* com similaridade maior que $0,6$ serão considerados *outliers* nessa base de dados.

Tabela 4 – Valores estatísticos das medidas de similaridade e distância das comparações entre palavras.

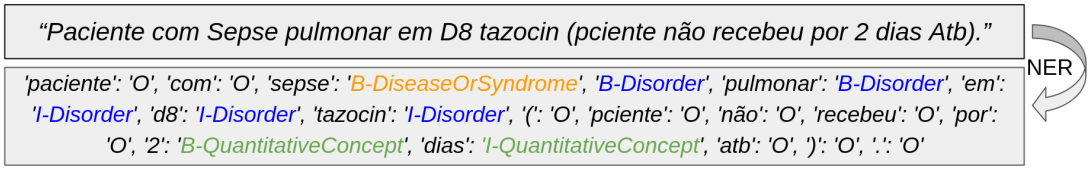
Estatísticas	**cosseno**	**Euclidiana**	**Manhattan**
#Palavras	6.767.520	6.767.520	6.767.520
Média	0,4844	21,5149	529,2993
Desvio padrão	0,1205	3,2700	78,1217
Mínimo	0,0291	0,0000	0,0000
25% (Q1)	0,4244	19,4941	480,9050
50% (mediana)	0,4925	21,2971	524,6183
75% (Q3)	0,5595	23,1582	569,9757
Máximo	1,0000	34,5744	865,5734

Fonte: o autor.

5.4 RECONHECIMENTO DE ENTIDADES NOMEADAS

Após a realização dos testes preliminares, iniciamos os testes com as ferramentas de NER. A primeira ferramenta utilizada e que estava sendo, até o momento, considerada como a principal ferramenta de NER da proposta foi o *BioBERT pt*. Escolhemos utilizar a versão *all* do modelo treinado. Ao usar o exemplo do autor, tivemos que fazer algumas alterações pois o exemplo faz uso de arquivos que não constam no diretório da biblioteca HuggingFace. A Figura 22 apresenta o resultado das entidades reconhecidas com as sentenças originais após os ajustes.

Figura 22 – Exemplo de entrada disponibilizada pelos desenvolvedores do *BioBERT pt*

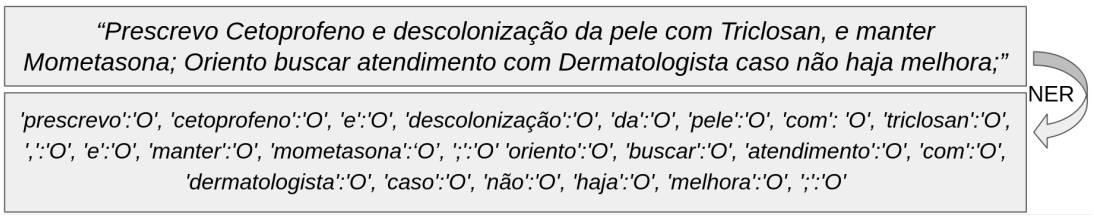


Fonte: o autor.

Com o exemplo funcionando, iniciamos a aplicação em nossos textos clínicos. Nesta etapa não houve reconhecimento de nenhuma entidade nos textos. Imaginamos haver algum problema com a implementação e tentamos utilizar a ferramenta de outras 2 formas. Primeiro baixando e configurando o *container* Docker disponibilizado pelo autor. O *container* executa conforme indicado, mas gera erro até mesmo para as frases de exemplo mostradas na tela. A segunda forma de usar a ferramenta foi criar um notebook que utiliza todos os 13 modelos

disponibilizados pelo autor no repositório do HuggingFace[6]. Esses modelos foram treinados separadamente para diferentes áreas médicas (e.g., Doenças, Procedimentos, Diagnostico, Química, Farmacêutica). Esta implementação passa cada documento por todos os 13 modelos verificando se alguma palavra é reconhecida por algum deles, o que deixou o processo muito lento. Cada palavra do documento recebe uma lista onde são concatenados os resultados de entidades reconhecidas. Isso é feito pois mais de 1 modelos pode reconhecer uma entidade para a palavra, como acontece para a palavra 'sepse' na Figura 22. Contudo, novamente não houveram resultados para nenhuma entidade reconhecida nos textos clínicos do nosso conjunto de dados. Um exemplo com um documento sendo passado isoladamente para os 13 modelos para um teste simples é ilustrado na Figura 23, onde a tag 'O' inserida após cada palavra significa que esta palavra está fora do vocabulário *"Out of Vocabulary"* do modelo.

Figura 23 – Exemplo de documento retirado dos Textos Clínicos



Fonte: o autor.

Para suprir a falta de entidades reconhecidas, recorremos ao uso de uma ferramenta de propósito geral: a DBpedia. Para usar sua ferramenta *web*, criamos uma rotina que faz consultas SPARQL ao seu *endpoint*, usando a biblioteca SPARQLWrapper do Python. Essas consultas foram feitas apenas para palavras alfanuméricas com tamanho maior que 2, para evitar fazer consultas com os artigos das sentenças. Usamos o padrão alfanumérico para fazer consultas com palavras como "COVID19" e não fazer consultas com caracteres especiais. Após o uso desse padrão e do limite de tamanho aplicado ficamos com um total de 1.001 palavras, sem repetições, para serem consultadas na DBpedia.

Para selecionar nos resultados das consultas realizadas as entidades da área clínica, foram anotados manualmente os tipos retornados para palavras da área clínica (e.g. remédios, doenças, procedimentos, etc...) mencionadas nos nossos textos clínicos. Em seguida, os tipos anotados foram divididos dentro das classes de interesse mencionadas na Seção 4.3. A Tabela 5 apresenta para cada classe de interesse os tipos anotados retornados pela DBPedia.

Tabela 5 – Tipos DBpedia separados por classes de interesse

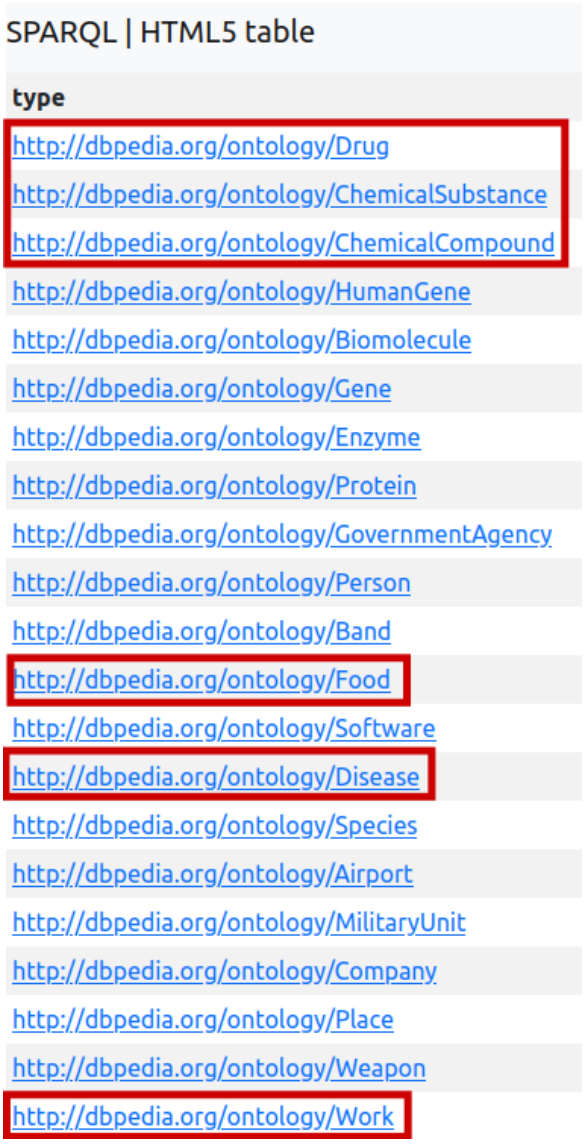
CHEM	**DISO**	**ANAT**	**PROC**	**SPEC**	**FOOD**
Drug	*Disease*	*Anatomical Structure*	*Work*	*Medical Specialty*	*Food*
Chemical Substance	-	-	-	*Person Function*	-
Chemical Compound	-	-	-	-	-

Fonte: o autor.

[6] Disponível em: https://huggingface.co/pucpr

Somente nos casos específicos em que uma palavra retorna tipos que a relacionam a mais de 1 das classes de interesse, é utilizado um terceiro filtro para pesquisar a palavra em inglês. Por exemplo, ao consultar a palavra "Mometasona" podemos observar na Figura 24 que o retorno contém tipos *Drug*, *Chemical Substance*, *Chemical Compound*, *Food*, *Disease* e *Work*.

Figura 24 – Retorno da consulta SPARQL feita na DBpedia para a palavra 'Mometasona'



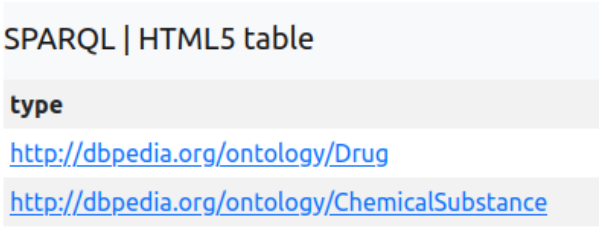
Fonte: o autor.

Esse retorno relaciona a palavra "Mometasona" a 4 classes de interesse. Nesse caso, usamos a biblioteca Python *deep-translator*[7], configurada para utilizar a *engine* do *Google-Translator*, para traduzir a palavra para o inglês (*Mometasone*) e concatenamos a tradução ao final da *string* "*http://dbpedia.org/resource/*". A *string* resultante da concatenação é utilizada para filtrar, dentre os resultados obtidos, os que possuam uma URI (do inglês, *Uniform Resource Identifier*) exatamente igual a ela. Obtemos assim uma correspondência exata da enti-

[7] https://deep-translator.readthedocs.io/en/latest/

dade desejada e verificamos novamente os tipos retornados. A Figura 25 apresenta o resultado da consulta da palavra “Mometasona” com o filtro da URI obtemos os tipos “*Drug*” e “*Chemical Substance*”. Dessa forma, a palavra “Mometasona” foi rotulada, pela função criada para classificação das entidades, com a classe de interesse CHEM.

Figura 25 – Retorno da consulta SPARQL feita na DBpedia para a palavra ’Mometasona’ usando o filtro por URI.



Fonte: o autor.

A Tabela 6 apresenta para cada classe de interesse as palavras reconhecidas como entidades. Podemos notar que algumas entidades são reconhecidas incorretamente, como a palavra “Reforço” que foi classificada como DISO e a palavra “Ainda” classificada como PROC. Mantivemos a classe SPEC fora da tabela por não ter ocorrência de entidade reconhecida dessa classe nos nossos experimentos. Esses resultados podem ter sido causados pelo fato da DBpedia ser uma ontologia de propósito geral, mantida por uma comunidade e que não conta com todas as ontologias médicas necessárias na língua portuguesa. Logo, ela contem alguns erros em suas base de dados e tem dificuldade de reconhecer entidades de áreas específicas. Usando o exemplo da palavra “Ainda”, ao realizar uma consulta por esta palavra no grafo de conhecimento da DBpedia, verificamos que a palavra corresponde ao nome de um álbum musical. Este tipo de entidade é classificada como “Work” pela DBpedia. Na Tabela 5 mostramos que o tipo “Work” é usado para classificar procedimentos médicos (PROC) em nossa tarefa de NER. Desta forma, a palavra “Ainda” está registrada na DBpedia com o mesmo tipo da palavra “Hemograma”. Isso faz nossa tarefa de NER classificar esta palavra como um procedimento médico.

Tabela 6 – Entidades identificadas separadas por classes de interesse

CHEM	DISO	ANAT	PROC	FOOD
Cetoprofeno	Febre	Mama	Ainda	Xarope
Bromexina	Dispneia	Narina	Partir	Vacina
Prednisona	Fome	Pele	Mental	-
Cortisol	Reforço	-	Paciente	-
Dexametasona	Dengue	-	Hemograma	-
Ipratrópio	Tosse	-	Oportunidades	-
Acetilcisteína	Coronavírus	-	Colateral	-
Pantoprazol	Dor	-	Mapa	-
Azitromicina	Gravidade	-	-	-

Salbutamol	Lesão	-	-	-
Amoxicilina	Diarreia	-	-	-
Aceclofenaco	Mialgia	-	-	-
Fluconazol	Hipotiroidismo	-	-	-
Desloratadina	Herpes	-	-	-
Paracetamol	Vírus	-	-	-
Água	Infecção	-	-	-
Loratadina	Sinusite	-	-	-
Simeticona	Anosmia	-	-	-
Ambroxol	Rinite	-	-	-
Naproxeno	Cefaleia	-	-	-
Ivermectina	Desidratação	-	-	-
Vitamina	Doença	-	-	-
Escitalopram	-	-	-	-
Baclofeno	-	-	-	-
Domperidona	-	-	-	-
Ciclobenzaprina	-	-	-	-
Itraconazol	-	-	-	-
Ibuprofeno	-	-	-	-
Nimesulida	-	-	-	-
Colesterol	-	-	-	-
Ureia	-	-	-	-
Creatinina	-	-	-	-
Sódio	-	-	-	-
Potássio	-	-	-	-
Triclosan	-	-	-	-
Mometasona	-	-	-	-
Fisioterapia	-	-	-	-
Propranolol	-	-	-	-
Bicarbonato	-	-	-	-
Topiramato	-	-	-	-

A Tabela 7 apresenta as estatísticas das entidades reconhecidas, separadas por classe de interesse, usando consultas SPARQL à DBpedia.

Tabela 7 – Estatísticas das entidades reconhecidas pela DBpedia em relação ao conjunto total de palavras (1.001).

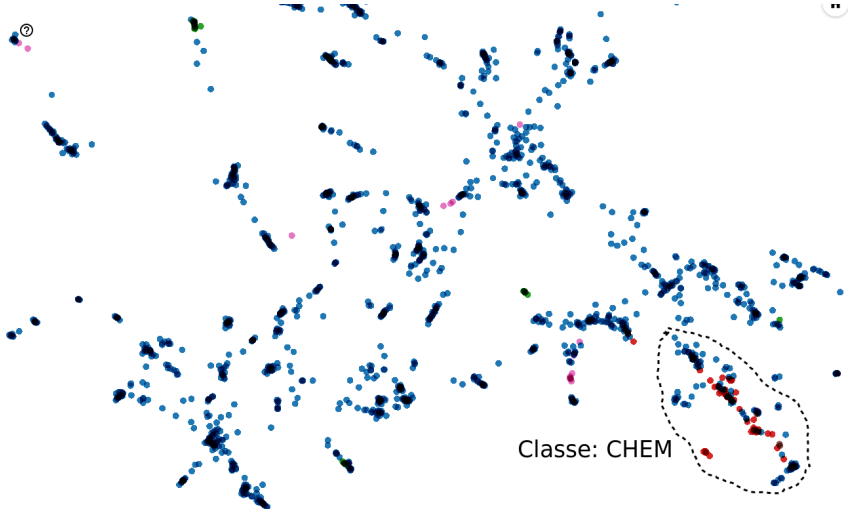
Classes	**CHEM**	**DISO**	**ANAT**	**PROC**	**FOOD**	**Total**
#Entidades reconhecidas	40	22	3	8	2	76
%Entidades reconhecidas por palavras consultadas	4,00%	2,20%	0,30%	0,80%	0,20%	7,59%

Fonte: o autor.

5.5 VISUALIZAÇÃO E MINERAÇÃO DE PADRÕES

Com os *embeddings* contextualizados das palavras gerados e as entidades reconhecidas mencionadas respectivamente classificadas nas classes de interesse, geramos as visualizações para complementar e dar suporte na mineração dos padrões. Criamos um experimento para analisar as classes morfo-sintáticas dos *embeddings* próximos às entidades reconhecidas. Nesse experimento criamos uma visualização no *Embeddings Projector* utilizando os *embeddings* de 10.690 palavras, gerados pela concatenação das 4 últimas camadas do *BERTimbau* (4.096 parâmetros por palavra). Usamos os 2 algoritmos de clusterização da ferramenta e o que apresentou um melhor resultado de agrupamento com relação às classes de interesse foi o UMAP. Na Figura 26 ilustramos o resultado obtido pelo UMAP, onde cada ponto representa a projeção em 2 dimensões destas palavras. As cores dos pontos distinguem as classes de interesse das palavras. Os pontos azuis representam as palavras comuns, não foram reconhecidas na tarefa de NER e consequentemente não pertencem a nenhuma classe de interesse. Em vermelho representam as entidades reconhecidas classificadas como CHEM, em rosa os classificados como DISO e em verde os classificados como PROC. As classes ANAT (azul-claro) e FOOD (marrom) apresentam poucos pontos (5 no total) e ficam imperceptíveis na figura. Para analisar as palavras próximas às classes de interesse, fizemos um recorte nos dados, na área de interesse em destaque na Figura 26. Essa área de interesse foi escolhida por ter um agrupamento de entidades reconhecidas classificadas como CHEM.

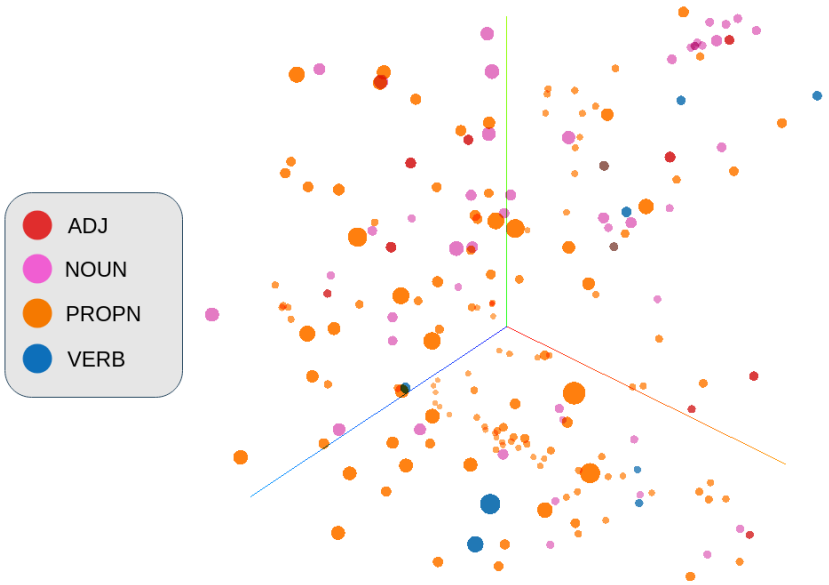
Figura 26 – Projeção 2D de agrupamentos UMAP de *embeddings* de 10.690 palavras.



Fonte: o autor.

Após o recorte nos dados obtivemos uma visualização com 202 pontos. Nesta visualização em três dimensões utilizando o algoritmo PCA configuramos a cor dos pontos para as suas respectivas PoS-*taggings*, com o intuito de observar possíveis verbos. A Figura 27 ilustra a visualização obtida com a respectiva legenda de cores. Nela é possível observar que a maioria dos pontos foi classificada pelo spaCy como substantivos (NOUN) e nomes próprios (PROPN). Logo, as entidades reconhecidas classificadas como medicamentos receberam *embeddings* contextualizados com valores próximos aos *embeddings* de palavras com semântica semelhante. Também é possível observar que existem poucos pontos (5) representando verbos. Ao fazer a verificação destes verbos, notamos que o spaCy classificou incorretamente os seguintes medicamentos como verbos: allegra, Dipirona, Loratadina e Azitromicina. Esta última inclusive é uma entidade reconhecida pela ferramenta de NER e classificada como CHEM. Os outros medicamentos citados não foram reconhecidas. Isso mostra que a tarefa de PoS-*tagging* realizada cometeu erros de classificação, mas os *embeddings* contextualizados desses medicamentos também receberam valores próximos aos *embeddings* de palavras semanticamente semelhantes.

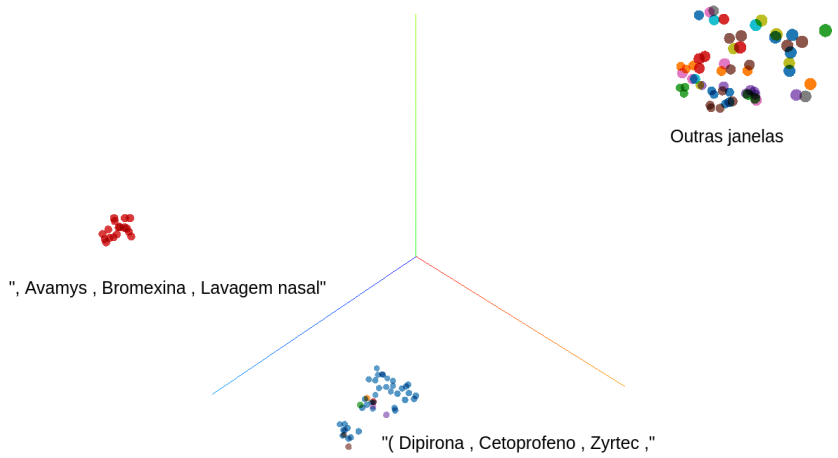
Figura 27 – Recorte feito na área de interesse da visualização dos *embeddings* de 10.690 palavras coloridas pela classe morfo-sintática



Fonte: o autor.

Também foram realizados experimentos com janelas centradas nas classes de interesse. Foram testadas janelas de tamanho 3, 5 e 7. Com isso, os tokens gerados pelo *BERTimbau* para as pontuações e caracteres especiais também são contabilizados para o tamanho da janela. Os *embeddings* contextualizados das janelas são obtidos através da técnica de *pooling MEAN* dos *embeddings* dos tokens da respectiva janela. Logo, com estes experimentos pretendíamos avaliar como os *embeddings* das palavras próximas às entidades reconhecidas mencionadas alteravam os *embeddings* destas entidades e se existia algum padrão observável com as entidades em contextos diferentes. Contudo, os nossos textos clínicos continham muito reuso de texto dificultando a mineração, uma vez que as palavras próximas das entidades reconhecidas mencionadas geralmente eram as mesmas. Como é possível observar na Figura 28, os *embeddings* das janelas centradas nas palavras “Cetoprofeno” (em azul) e “Bromexina” (em vermelho) criaram 2 agrupamentos separados dos demais. O último agrupamento visível contém o restante das janelas ao redor das outras entidades reconhecidas, e não foi possível identificar o motivo do agrupamento.

Figura 28 – Projeção 3D de agrupamentos UMAP dos *embeddings* consolidados de 122 janelas de tamanho 3.



Fonte: o autor.

Para verificar os padrões contidos nos textos clínicos, usamos o Algoritmo 3 para obter as palavras mais similares às entidades reconhecidas mencionadas. Filtramos as comparações realizadas utilizando o *threshold* de $0,8$, usado por Sorato, Goularte e Fileto (2020). A Tabela 8 mostra as quantidades de entidades comparadas, separadas por classes de interesse, assim como as médias das medidas de distâncias e similaridade obtidas.

Tabela 8 – Quantidades e médias das distâncias e similaridade de entidades nomeadas com *threshold* maior ou igual $0,8$.

Classe	Qtd.	Média cos	Média Euc	Média Man
CHEM	896	0.828	10.8013	267.8409
DISO	14	0.8068	12.5976	315.3274
PROC	106	0.9846	3.7027	94.8284

Fonte: o autor.

Na Tabela 9 podemos visualizar as entidades reconhecidas mencionadas nos textos clínicos com a maior quantidade de vezes que ela teve a similaridade dos seus *embeddings* acima do threshold utilizado. A primeira entidade da lista (Cetoprofeno) tem 132 ocorrências de palavras similares a ela.

Tabela 9 – Lista das 10 palavras/entidades com maior quantidade de ocorrências de similaridades dos *embeddings*, classes, sentenças e quantidades.

Entidade Reconhecida	**Classe**	**Sentença**	**Qtd.**
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Zyrtec , Avamys , Bromexina , Lavagem nasal com SF) ;	132
Paracetamol	CHEM	Dip 1 g 6/6h , SN + **Paracetamol** 750	76
ainda	PROC	Oriento **ainda** a retornar o contato em caso de piora dos sintomas ;	50
Cetoprofeno	CHEM	Conduta : Prescrevo **Cetoprofeno** e descolonização da pele com Triclosan , e manter Mometasona ;	48
Desloratadina	CHEM	**Desloratadina** 0,5 mg ml 10 ml dia , se tosse seca	42
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Baclofeno , Tylex) ;	40
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Zyrtec) ;	40
ibuprofeno	CHEM	Substituo **ibuprofeno** por 3 dias de cetoprofeno 100 mg 12/12h associado a dipirona em dose analgésica (1 g) .	38
cetoprofeno	CHEM	Substituo ibuprofeno por 3 dias de **cetoprofeno** 100 mg 12/12h associado a dipirona em dose analgésica (1 g) .	38
Mometasona	CHEM	Conduta : Prescrevo Cetoprofeno e descolonização da pele com Triclosan , e manter **Mometasona** ;	36

Fonte: o autor.

A Tabela 10 lista essas 132 ocorrências agrupadas por palavras iguais. Podemos notar que mesmo comparando com todas as palavras, as maiores similaridades ocorrem com outros medicamentos.

Tabela 10 – Lista das 132 palavras com *embeddings* similares à Entidade Reconhecida 'Cetoprofeno'

Palavra	**Qtd.**
Dipirona	36
Cetoprofeno	30
Paracetamol	18
Bromexina	18
cetoprofeno	12
Baclofeno	6
ibuprofeno	6
Ciprofloxacino	6

Fonte: o autor.

A Tabela 11 lista as 5 janelas de sentenças com maior número de ocorrências de entidades da classe CHEM. Estas são janelas das sentenças relevantes (com entidades reconhecidas) com maior número de outras janelas similares também relevantes. A primeira janela da lista tem 66 ocorrências de janelas similares a ela.

Tabela 11 – Lista das 5 janelas de tamanho 3 com *embeddings* com maior similaridade

Janela	Sentença	Qtd.
“, Bromexina ,”	Prescrevo sintomáticos SN (Dipirona , Cetoprofeno , Zyrtec , Avamys **, Bromexina ,** Lavagem nasal com SF) ;	66
“, Cetoprofeno ,”	Prescrevo sintomáticos SN (Dipirona **, Cetoprofeno ,** Zyrtec , Avamys , Bromexina , Lavagem nasal com SF) ;	54
“com Paracetamol ,”	Prescrevo sintomáticos SN (Dipirona intercalando **com Paracetamol ,** Celerg , Lavagem nasal com SF) ;	22
“, Baclofeno ,”	Prescrevo sintomáticos SN (Dipirona , Cetoprofeno **, Baclofeno ,** Tylex) ;	20
“com Triclosan ,”	Conduta : Prescrevo Cetoprofeno e descolonização da pele **com Triclosan ,** e manter Mometasona ;	20

Fonte: o autor.

A Tabela 12 lista essas 66 ocorrências agrupadas por janelas iguais. Podemos notar que novamente, mesmo comparando com janelas centradas em todas as entidades, as maiores similaridades ocorrem com janelas centradas em outros medicamentos.

Tabela 12 – Lista das 66 janelas mais similares da janela “, Bromexina ,”.

Janela	Sentença	Qtd.
, Cetoprofeno ,	Prescrevo sintomáticos SN (Dipirona **, Cetoprofeno ,** Zyrtec , Avamys , Bromexina , Lavagem nasal com SF) ;	18
, Bromexina ,	Prescrevo sintomáticos SN (Dipirona , Cetoprofeno , Zyrtec , Avamys **, Bromexina ,** Lavagem nasal com SF) ;	12
, Baclofeno ,	Prescrevo sintomáticos SN (Dipirona , Cetoprofeno **, Baclofeno ,** Tylex) ;	6
, Cetoprofeno ,	Prescrevo sintomáticos SN (Dipirona **, Cetoprofeno ,** Zyrtec) ;	6
, Cetoprofeno ,	Prescrevo sintomáticos SN (Dipirona **, Cetoprofeno ,** Baclofeno , Tylex) ;	6
com Paracetamol ,	Prescrevo sintomáticos SN (Dipirona intercalando **com Paracetamol ,** Celerg , Lavagem nasal com SF) ;	6
, dispneia ,	Oriento à buscar atendimento de urgência em caso de sintomas graves (febre >39,5ºC que não cessa com uso de antitérmico **, dispneia ,** desmaio e queda do quadro geral) ;	6
com Triclosan ,	Conduta : Prescrevo Cetoprofeno e descolonização da pele **com Triclosan ,** e manter Mometasona ;	6

Fonte: o autor.

5.6 DISCUSSÃO

No decorrer no trabalho notamos que o uso de *embeddings* para representação de palavras não foi muito eficiente em demonstrar os padrões que pretendíamos minerar. Isso pode ter sido causado pelos valores de *embeddings* gerados para os substantivos das sentenças serem próximos uns dos outros, mas distantes dos verbos usados nas sentenças. Já os *embeddings* gerados para os verbos não apresentaram agrupamentos nas visualizações realizadas.

Os resultados ruins dos experimentos de NER com o modelo *BioBERT pt* (em termos de cobertura e precisão) podem se devidos ao fato do software estar mal documentado, atrapalhando o uso correto de seus modelos. Dessa forma não conseguimos reconhecer corretamente todas as entidades mencionadas mesmo para um conjunto de dados pequeno. O conjunto de dados disponibilizado, além de ser pequeno, continha muito reuso de sentenças e muitos caracteres especiais, diminuindo a eficácia do PLN. Também não tivemos acesso a um corpus padrão ouro para realizar uma métrica de acurácia das ferramentas de NER.

A tarefa de visualizar e selecionar *embeddings* (descrita na Seção 4.5) possibilitou identificar alguns agrupamentos. Principalmente as visualizações de *embeddings* de algumas

classes de interesse mostraram agrupamentos. Por exemplo, como o apresentado na Figura 26, o agrupamento das palavras em torno de entidades reconhecidas da classe de interesse CHEM é bem definido. 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 *embeddings* da concatenação das 4 últimas camadas do *BERTimbau*, de acordo com a recomendação do Devlin et al. (2019), que gera *embeddings* com 4.096 valores (dimensões). Essa escolha fez com que ocasionalmente as estruturas criadas nos experimentos consumissem todos os 12 GB de memória disponibilizados. 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 clínicos visando dar suporte a classificação não supervisionada de tais textos.

Realizamos a mineração de padrões usando medidas de distância e similaridade entre os *embeddings* de palavras, assim como usando casamento de classes de interesse das entidades reconhecidas nos textos.

O estudo e compreensão dos métodos, técnicas e ferramentas de PLN permitiu desenvolver algoritmos para mineração de padrões morfo-semânticos em textos clínicos. Os resultados dos algoritmos demostram que o BERT usa o contexto dos documentos para gerar os *embeddings* dos medicamentos próximos aos *embeddings* de outras palavras mencionadas nos mesmos contextos textuais, tais como doenças tratadas com os respectivos medicamentos. Isso não permite discriminar medicamentos e doenças, por exemplo, em grupos distintos de *embeddings*. Também observamos que os *embeddings* gerados para os verbos geralmente usados pelos profissionais da saúde para fazer prescrições aos pacientes não são necessariamente próximos dos *embeddings* dos medicamentos prescritos.

Os algoritmos de mineração de padrões morfo-semânticos em textos clínicos precisam ser reavaliados utilizando uma base de dados padrão ouro. Um padrão ouro possibilitaria uma análise quantitativa dos resultados, pois sem ele apenas a análise qualitativa dos padrões foi realizada.

Trabalhos futuros incluem: (i) aprimoramento dos algoritmos para mineração de padrões morfo-semânticos baseados em proximidade de *embeddings* e classes de interesse obtidas via NER; (ii) comparação mais exaustiva desses algoritmos, usando bases de dados maiores e mais diversificadas; (iii) desenvolvimento e uso de uma ferramentas de NER específicas para textos clínicos; (iv) utilizar modelos de *word embeddings* mais recentes como *BLOOM*[1] e (v) investigação das possibilidades de aplicação da abordagem proposta a textos de outros domínios, tais como textos jurídicos, científicos e de literatura.

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

REFERÊNCIAS

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.

BENÍCIO, D. H. P. **Aplicação de mineração de texto e processamento de linguagem natural em prontuários eletrônicos de pacientes para extração e transformação de texto em dado estruturado**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2020.

BERTALAN, V. G. F.; RUIZ, E. E. S. Predicting judicial outcomes in the brazilian legal system using textual features. In: **DHandNLP@ PROPOR**. [S.l.: s.n.], 2020. p. 22–32.

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.

CAMPILLOS-LLANOS, L. et al. A clinical trials corpus annotated with umls entities to enhance the access to evidence-based medicine. **BMC medical informatics and decision making**, BioMed Central, v. 21, n. 1, p. 1–19, 2021.

CEGALLA, D. P. Novíssima gramática da língua portuguesa. Companhia Ed. Nacional, 2008.

CORDEIRO, B. C. **BERT e Word2Vec: Uma análise inferencial e computacional na classificação 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.

EHRLINGER, L.; WÖSS, W. Towards a definition of knowledge graphs. **SEMANTiCS (Posters, Demos, SuCCESS)**, v. 48, n. 1-4, p. 2, 2016.

FELLBAUM, C. (Ed.). **WordNet: An Electronic Lexical Database**. Cambridge, MA: MIT Press, 1998. (Language, Speech, and Communication). ISBN 978-0-262-06197-1.

GALVÃO, M. C. B.; RICARTE, I. L. M. O prontuário eletrônico do paciente no século xxi: contribuições necessárias da ciência da informação. **InCID: Revista de Ciência da Informação e Documentação**, v. 2, n. 2, 2011.

GLIGIC, L. et al. Named entity recognition in electronic health records using transfer learning bootstrapped neural networks. **Neural Networks**, v. 121, p. 132–139, 2020. ISSN 0893-6080. Disponível em: https://www.sciencedirect.com/science/article/pii/S089360801930259X.

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.

LEE, J. et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, Oxford University Press (OUP), sep 2019. Disponível em: https://doi.org/10.1093%2Fbioinformatics%2Fbtz682.

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.

LI, J. et al. A survey on deep learning for named entity recognition. **IEEE Transactions on Knowledge and Data Engineering**, IEEE Computer Society, Los Alamitos, CA, USA, v. 34, n. 01, p. 50–70, jan 2022. ISSN 1558-2191.

LIN, Y. et al. Learning entity and relation embeddings for knowledge graph completion. In: **Twenty-ninth AAAI conference on artificial intelligence**. [S.l.: s.n.], 2015.

LOPES, A. A. Prontuário orientado por problemas e evidências (pope). 2020.

LY, A.; UTHAYASOORIYAR, B.; WANG, T. A survey on natural language processing (nlp) and applications in insurance. **arXiv preprint arXiv:2010.00462**, 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.

MARTINS, L. J. Cálculo de indicadores financeiros com auxílio do processamento de linguagem natural. Florianópolis, SC., 2021.

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 preprint arXiv:1301.3781**, 2013.

MORO, A.; RAGANATO, A.; NAVIGLI, R. Entity linking meets word sense disambiguation: a unified approach. **Transactions of the Association for Computational Linguistics**, MIT Press, Cambridge, MA, v. 2, p. 231–244, 2014. Disponível em: https://aclanthology.org/Q14-1019.

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.

NAVIGLI, R. et al. Ten years of babelnet: A survey. In: ZHOU, Z.-H. (Ed.). **Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21**. International Joint Conferences on Artificial Intelligence Organization, 2021. p. 4559–4567. Survey Track. Disponível em: https://doi.org/10.24963/ijcai.2021/620.

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

PINTO, V. B.; SALES, O. M. M. Proposta de aplicabilidade da preservação digital ao prontuário eletrônico do paciente. **RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação**, v. 15, n. 2, p. 489–507, 2017.

PIRIS, Y.; GAY, A.-C. Customer satisfaction and natural language processing. **Journal of Business Research**, Elsevier, v. 124, p. 264–271, 2021.

QI, P. et al. Stanza: A python natural language processing toolkit for many human languages. **arXiv preprint arXiv:2003.07082**, 2020.

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.

RIDGWAY, J. P. et al. Natural language processing of clinical notes to identify mental illness and substance use among people living with hiv: retrospective cohort study. **JMIR Medical Informatics**, JMIR Publications Inc., Toronto, Canada, v. 9, n. 3, p. e23456, 2021.

SANTOS, Y. J. A. d. Avaliação de técnicas de aprendizado supervisionado no auxílio a diagnósticos médicos. 2021.

SCHNEIDER, E. T. R. et al. Biobertpt-a portuguese neural language model for clinical named entity recognition. In: **Proceedings of the 3rd Clinical Natural Language Processing Workshop**. [S.l.: s.n.], 2020. p. 65–72.

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.

SORIN, V. et al. Deep learning for natural language processing in radiology—fundamentals and a systematic review. **Journal of the American College of Radiology**, Elsevier, v. 17, n. 5, p. 639–648, 2020.

SOUZA, A. D. de; FELIPE, E. R. Processamento de linguagem natural aplicado à anamneses do domínio da ginecologia. 2021.

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.

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.

WANG, J. et al. Systematic evaluation of research progress on natural language processing in medicine over the past 20 years: Bibliometric study on pubmed. **Journal of medical internet research**, JMIR Publications Inc., Toronto, Canada, v. 22, n. 1, p. e16816, 2020.

WANG, Y.; GUO, M. A short analysis of discourse coherence. **Journal of Language Teaching and Research**, Citeseer, v. 5, n. 2, p. 460, 2014.

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.

WEED, L. L. Medical records that guide and teach. **New England Journal of Medicine**, Mass Medical Soc, v. 278, n. 12, p. 652–657, 1968.

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-.

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.

WU, Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. **arXiv preprint arXiv:1609.08144**, 2016.

XIAN, Y. et al. Neural-symbolic reasoning over knowledge graph for multi-stage explainable recommendation. **arXiv preprint arXiv:2007.13207**, 2020.

XIE, L. et al. Explainable recommendation based on knowledge graph and multi-objective optimization. **Complex & Intelligent Systems**, Springer, v. 7, n. 3, p. 1241–1252, 2021.

ZHANG, Z.; ZHAO, H.; WANG, R. Machine reading comprehension: The role of contextualized language models and beyond. **arXiv preprint arXiv:2005.06249**, abs/2005.06249, n. 1, 2020.

APÊNDICE A – ARTIGO DO TRABALHO

Mineração de padrões morfo-semânticos em textos clínicos

Nathan Cezar Cardoso[1], **Osmar de Oliveira Braz Junior**[1], **Renato Fileto**[1]

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

nathan.cezar@grad.ufsc.br, osmar.braz@posgrad.ufsc.br

renato.fileto@inf.ufsc.br

Abstract. *Currently, large volumes of texts from different domains have been collected daily on digital platforms. Several tools for Natural Language Processing (NLP), text mining and data science allow extracting information, analyzing and determining certain texts according to their contents. This work developed and evaluated techniques and algorithms to mine morpho-semantic patterns in clinical texts, with the aim of automating the classification and screening of these texts and enabling analysis of their contents. The results were quantitatively evaluated, in terms of the distribution of instances of the patterns mined in the document collections.*

Keywords: *Text mining, Unsupervised classification of texts, Natural Language Processing, Embeddings, Entity recognition. Clinical Texts.*

Resumo. *Atualmente, enormes volumes de textos de diversos domínios têm sido coletados diariamente em plataformas digitais. Várias ferramentas para Processamento de Linguagem Natural (PLN), mineração de textos e ciência de dados permitem extrair informação, analisar e classificar certos textos conforme os seus conteúdos. Este trabalho desenvolveu e avaliou técnicas e algoritmos para minerar padrões morfo-semânticos em textos clínicos, com o intuito de automatizar a classificação e a triagem desses textos e possibilitar análises de seus conteúdos. Os resultados foram avaliados quantitativamente, em termos da distribuição de instâncias dos padrões minerados nas coleções de documentos.*

Palavras-chave: *Mineração de textos, Classificação não-supervisionada de textos, Processamento de Linguagem Natural, Embeddings, Reconhecimento de entidades, Textos Clínicos.*

1. Introdução

O PLN está atualmente consagrado na ciência da computação. Novas aplicações aparecem a todo momento, auxiliando, por exemplo, no entendimento da experiência e satisfação do cliente [Piris and Gay 2021] ou na previsão de resultados judiciais [Bertalan and Ruiz 2020]. Tarefas de PLN como classificação morfossintática de palavras (do inglês, POS-*Tagging*), juntamente com *embeddings* de palavras e aprendizado de máquina, permitem também detectar e analisar discursos de ódio em textos de mídias sociais [Sorato et al. 2020, Leite et al. 2020], entre várias outras aplicações. No contexto de saúde, o uso de técnicas e ferramentas de PLN em aplicações também vem aumentando consideravelmente nos últimos anos, sobretudo em resposta à pandemia do COVID-19.

Abordagens atuais utilizam técnicas e ferramentas de PLN e mineração de textos (do inglês, TM) em diferentes áreas da medicina. Por exemplo, no reconhecimento de entidades nomeadas (do inglês, NER) [Li et al. 2022] para extração de termos técnicos de anamneses (registro completo da história clínica de um paciente) de prontuários eletrônicos do domínio da ginecologia [de Souza and Felipe 2021], usando PLN em anotações clínicas para detectar doenças mentais e uso de substâncias entre pessoas vivendo com HIV [Ridgway et al. 2021] e utilizando PLN e aprendizado profundo (do inglês, DL) na área da radiologia [Sorin et al. 2020]. Todavia, ainda há vários desafios em aberto envolvendo PLN no domínio da saúde.

1.1. Descrição do Problema

Com o aumento dos tele-atendimentos e extensa digitalização dos dados clínicos (colhidos em atendimentos), cresce cada vez mais a necessidade de se minerar esses dados. Dados (semi)-estruturados, informação e conhecimento extraídos de dados clínicos colhidos na forma de textos em linguagem natural podem auxiliar e aprimorar sistemas de vigilância de saúde (auxiliando na análise de tendências, rápida identificação de doenças e surtos, além de fatores de risco). Isso pode contribuir para a criação de melhores estratégias de prevenção de doenças e resposta às tendências detectadas ou mesmo crises, entre outras possibilidades.

O problema tratado neste trabalho é minerar certos padrões de linguagem, a que denominamos morfo-semânticos, em textos clínicos. Um padrão morfo-semântico [Goularte et al. 2020] é caracterizado pela presença de palavras com classes gramaticais equivalentes ou compatíveis e sentidos similares em diferentes instâncias de textos (e.g., sentenças, documentos curtos).

Ferramentas de PLN já disponíveis podem ser usadas para efetuar não apenas a classificação morfossintática de palavras (POS-*Tagging*) para identificar verbos, substantivos, etc. Elas podem ser usadas também para o reconhecimento de entidades nomeadas (NER) e classificá-las em categorias como alimentos, medicamentos e especialidades médicas.

Uma ocorrência de verbo no texto pode estar associada a várias entidades, sendo as palavras que denotam tanto o verbo quanto as entidades, compatíveis em termos de classe morfossintática (verbo com verbo, substantivo com substantivo) e sentido (similar ou referente à mesma categoria de entidades) com outra(s) palavras em trechos distintos de textos. Por isso denominamos tais padrões morfo-semânticos. Todavia, vale salientar que para um conjunto de trechos de texto serem instâncias do mesmo padrão morfo-semântico, não precisa necessariamente haver correspondência biunívoca entre suas palavras. Basta haver alguns casamentos de palavras, em termos das respectivas classes morfo-sintáticas e sentidos. Por exemplo, a sentença 2 e a primeira oração da sentença 3, podem ser consideradas instâncias de um mesmo padrão morfo-semânticos, pois ambas fazem prescrição de medicamento(s). Por outro lado, a segunda oração da sentença conjugada 3 e a sentença 4 são instâncias de outro padrão, pois ambas fazem encaminhamento a especialidade médica.

1.2. Objetivos

O objetivo 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, foram realizados estudos sobre o estado da arte em tarefas e ferramentas de PLN que podem ser usadas na preparação dos textos clínicos a serem minerados, nos próprios algoritmos de mineração de padrões morfo-semânticos e na avaliação dos seus resultados. Tais tarefas incluíram normalização de texto (tokenização, stemming, etc.), classificação morfossintática (POS-*Tagging*), reconhecimento de entidades nomeadas (NER) e cálculo de similaridade entre *embeddings* de palavras. Posteriormente, foram 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 tiveram objetivo de preparação adequada dos dados e definição de parâmetros para mineração de padrões morfo-semânticos, incluindo, entre outros, funções de similaridade ou distância semântica e patamares (*thresholds*) a serem utilizadas na mineração.

Para resolver o problema de minerar automaticamente padrões morfo-semânticos podem ser utilizadas diversas alternativas de medidas de similaridade e outros critérios para determinar compatibilidade de palavras. Neste trabalho exploramos *embeddings* contextualizados de palavras e categorias de entidades identificadas mediante aplicação de NER para determinar compatibilidade semântica de palavras, além de seus *POS-Tags* (e.g., substantivo, verbo), nos algoritmos que desenvolvemos, adaptamos e avaliamos para minerar automaticamente os padrões morfo-semânticos existentes nos textos clínicos escritos por profissionais da saúde durante os atendimentos médicos.

1.4. Estrutura do trabalho

O restante deste trabalho está estruturado como se segue. O Seção 2 descreve os fundamentos utilizados no trabalho e necessários ao seu entendimento. O Seção 3 discute os trabalhos relacionados e compara as características do trabalho aqui proposto com o estado-da-arte. O Seção 4 descreve a proposta para minerar padrões em textos clínicos usando *embeddings* contextualizados. Finalmente o Seção 5 delineia o plano de experimentos para avaliar a proposta e reporta experimentos iniciais.

2. Fundamentos

Esta seção descreve brevemente os principais conceitos e técnicas usados neste trabalho para a mineração de padrões morfo-semânticos em textos clínicos. Primeiramente,

a seção 2.1 fornece uma visão geral das características de textos clínicos e do padrão (SOAP), o qual inclui um protocolo utilizado para coletar dados clínicos em forma textual e um formato para representá-los e armazená-los. Posteriormente, a seção 2.2 descreve as tarefas de Processamento de Linguagem Natural (PLN) usadas nas soluções investigadas neste trabalho. Finalmente, a seção 2.3 apresenta uma breve introdução a *embeddings*, ao modelo contextualizado de linguagem BERT, empregado para gerar os *embeddings* usados neste trabalho, e ao cálculo de funções de distância e similaridade entre *embeddings*.

2.1. Textos clínicos

Os prontuários são documentos legais que devem conter todos os dados assistenciais prestados aos pacientes, sejam registros de internação ou consultas em consultório, tornando-se um documento importante para a integração do cuidado. Além disso, apresenta informações sociais, demográficas e socioeconômicas.

Os dados clínicos são considerados informações para monitorar a saúde do indivíduo, obtidas conforme o paciente é observado: exames de imagem e laboratoriais, histórico médico, evolução, prescrição, sinais vitais e avaliação de risco e resumo de alta. Por outro lado, as informações não clínicas estão relacionadas a questões administrativas, não as condições de saúde. São dados relacionados às atividades nutricionais, farmácia, manutenção, aplicação de protocolo e materiais consumidos [Pinto and Sales 2017].

Profissionais de saúde debatem sobre como deve ser organizado um prontuário médico antes mesmo da criação dos sistemas informatizados. Na década de 1960 foi proposto um modelo de prontuário [Weed 1968] que é atualmente adotado em diversos centros de saúde de todo o mundo [Lopes 2020]. O modelo de prontuário proposto foi denominado POPE (do inglês, *Problem-Oriented Record* - POR) e tem como pontos mais relevantes a Lista de Problemas e as Notas de Evolução no Modelo **SOAP**. Cada letra da sigla do modelo SOAP se refere a um dos quatro aspectos fundamentais das notas de evolução: **dados subjetivos (S), dados objetivos (O), avaliação (A) e planos (P)** [Lopes 2020]. No campo subjetivo (S) o profissional registra os relatos do paciente documentando os sinais e sintomas mencionados, descrevendo o motivo que o trouxe à consulta. No campo objetivo (O), é feito o registro dos dados observáveis e mensuráveis avaliados pelo profissional da saúde, abrangendo o exame clínico e os exames complementares. No campo avaliação (A) o profissional descreve a impressão/interpretação que infere a partir das queixas subjetivas (S) do paciente e dos achados da parte objetiva (O). Neste campo é descrita a lista de problemas da consulta atual. Por último, no campo plano (P) é elaborada uma proposta de abordagem planejada para os problemas levantados. Esta proposta pode compreender medicações prescritas, solicitações de exames complementares, orientações realizadas, encaminhamentos e pendências para o próximo atendimento.

Portanto, a concentração de informação reunida por profissionais de saúde neste documento pode ser usada para formular uma síntese automatizada de dados úteis com base nas necessidades desses profissionais e da gestão das unidades de saúde. Desta forma, apoiando uma melhor tomada de decisão para resolver os casos.

2.2. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) refere-se à capacidade das máquinas de entender e explicar a maneira como os humanos escrevem e falam [Wang et al. 2020].

Envolve o estudo de várias teorias e métodos que podem realizar uma comunicação eficaz entre humanos e computadores em linguagem natural e é uma direção importante no campo da inteligência artificial.

Para processar um texto escrito em uma linguagem desenvolvida por humanos, é necessário seguir uma série de etapas iterativas envolvendo o PLN. Essas etapas incluem pré-processamento dos dados; representação de palavras como vetores numéricos; treinamento de um modelo de Machine Learning (ML) usando a representação da etapa anterior; avaliação do modelo. No decorrer desta sessão essas etapas serão aprofundadas.

Pré-processamento

O PLN pode se tornar uma tarefa árdua de realizar se o formato dos dados não for padronizado. Nesse sentido, alguns passos que podem ser realizados nos textos para tornar o processamento de texto mais simples e menos impreciso [Ly et al. 2020]:

- Carregar o texto: carregar o texto na memória, corretamente, lidando com a codificação desejada;
- *Lowercasing*: operação de converter todas as letras para minúsculas, isso evita que a mesma palavra seja identificada como duas devido às letrar maiúsculas;
- Remoção de *stopwords*: remover palavras menos significativas como preposições, conjunções, etc. porque elas não carregam nenhum significado e não são uteis para a análise;
- Processamento de caracteres especiais: dependendo do contexto, caracteres especiais como interrogações ou exclamações são importantes para o processamento posterior. Contudo, é necessário remover duplicidades ou caracteres indesejados;
- Tokenização: dividir o texto em um conjunto de unidades menores (tokens). Essas unidades podem ser sentenças ou até mesmo palavras.

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.

Reconhecimento de Entidades Nomeadas

O reconhecimento de entidades nomeadas (do inglês *Named Entity Recognition - NER*) [Li et al. 2022] é uma das mais importantes tarefas para extração de informação em tex-

tos. Consiste em reconhecer, delimitar e classificar o sentido de menções em texto livre a entidades nomeadas (instâncias) de classes como pessoas, instituições, lugares, data e hora. Diversas ferramentas de PLN oferecem soluções para NER, usualmente baseadas em modelos de aprendizado de máquina hoje em dia [Speck and Ngonga Ngomo 2014, de Abreu et al. 2017]. Tais ferramentas costumam integrar NER a reconhecimento de conceitos (e.g., medicamentos, componentes químicos, comidas, partes da anatomia humana, procedimentos e especialidades médicas) e por vezes também à tarefa de ligação de entidades (do inglês *Entity Linking - EL*) [Oliveira et al. 2021, Sevgili et al. 2022] que liga cada menção reconhecida em um texto à descrição da entidade à qual a menção se refere, através de um ponteiro para a respectiva descrição em alguma base de dados (e.g. Wikipedia) ou de conhecimento tal como um grafo de conhecimento (e.g. DBpedia [Lehmann et al. 2015]).

Outra possibilidade é usar as atuais soluções para NER baseadas em modelos contextualizados de linguagem treinados com textos da área de saúde e biologia em língua portuguesa, tais como o $BioBERTpt$, que também teve sintonia fina para efetuar NER em tais tipos de textos.

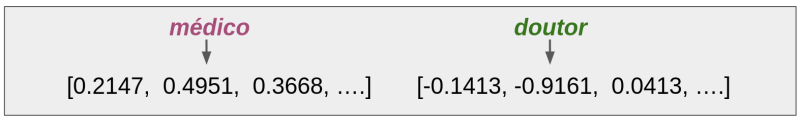
As ferramentas de NER podem ser usadas, por exemplo, na implementação em sistemas de busca e extração de informação. Estes sistemas auxiliam profissionais da área médica a navegar por palavras chave em publicações ao realizar buscas por seus temas de interesse. O acesso a tipos específicos de publicações poderia ser mais rápido se os profissionais pudessem customizar sua busca e restringi-la às classes semânticas escolhidas [Campillos-Llanos et al. 2021].

Em seu artigo, o autor separa as entidades em 4 classes, denominadas Grupos Semânticos, do Sistema Médico Unificado de Linguagem (do inglês, *Unified Medical Language System* - UMLS): entidades relativas a patologias (DISO), entidades anatômicas (ANAT), substâncias bioquímicas ou farmacológicas (CHEM) e procedimentos diagnósticos ou terapêuticos e exames laboratoriais (PROC).

2.3. 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: “médico” e “doutor”.

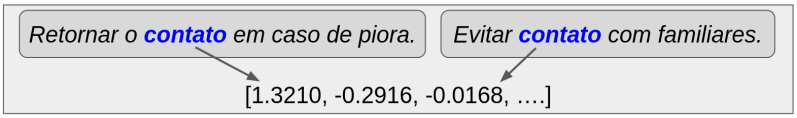
Figura 1. *Embeddings* das palavras



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 seu significado ou mesmo classe morfossintática mude em diferentes contextos onde é usada, como ilustrado na Figura 2.

Figura 2. *Embedding* estáticos de uma mesma palavra em contextos diferentes



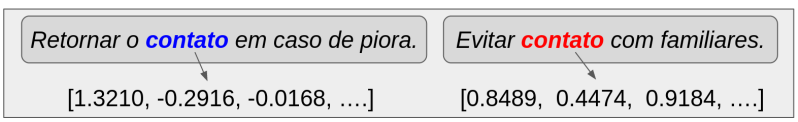
Na Figura 2 é possível ver que a palavra **"contato"** tem sentidos distintos nas duas sentenças. Na sentença da esquerda, tal palavra faz o papel de substantivo e tem o sentido de comunicar, enquanto na sentença da direta tem o sentido de toque/proximidade. Assim, as nuances de significado não são capturadas 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.

2.4. Modelos Contextualizados de Linguagem

Estudos recentes têm buscado meios de automatizar a compreensão da comunicação em linguagem natural através da aplicação de MCL (do inglês, *Contextualized Language Model*) [Zhang et al. 2020]. Um MCL como o BERT captura várias características da linguagem, incluindo nuances de significados que variam de acordo com o uso das palavras [Devlin et al. 2019], gerando *embeddings* ajustados a cada contexto textual onde ocorre um mesmo léxico. O BERT tem propiciado ganhos consideráveis de desempenho em diversas tarefas de PLN [Devlin et al. 2019], tais como segmentação [Muller et al. 2019] e comparação semântica de sentenças [Reimers and Gurevych 2019] e classificação de documentos [Ostendorff et al. 2020].

Atuais MCLs como o BERT [Devlin et al. 2019] utilizam vetores derivados de uma rede bidirecional e conseguem capturar o contexto em que uma palavra ocorre em um texto, capturando assim o seu significado específico naquele contexto, como ilustrado pela Figura 3.

Figura 3. *Embeddings* contextualizados de uma mesma palavra em contextos diferentes



BERT: *Bidirectional Encoder Representations from Transformers*

O BERT (acrônimo do inglês, *Bidirectional Encoder Representations from Transformers*) [Devlin et al. 2019], é uma rede neural profunda projetada para treinar modelos de linguagem através do processamento bidirecional de documentos não rotulados, considerando os contextos das palavras em ambas as direções. Por se tratar de um modelo flexível,

o resultado do pré-treinamento pode sofrer ajustes finos (do inglês, *Fine-Tuning*) com a adição de uma camada de saída, para criação de novos modelos ajustados para realizar tarefas *downstream*, sem a necessidade de modificações significativas na sua arquitetura. Desta forma, o BERT é capaz de lidar com uma variedade de tarefas, incluindo QA, classificação e inferências sobre linguagem [Devlin et al. 2019].

O BERT original vem pré-treinado com um corpus genérico formado por documentos não rotulados, o qual inclui a Wikipedia (2,5 bilhões de palavras) e o Toronto Book Corpus (800 milhões de palavras), ambos em língua inglesa. 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 nos textos clínicos. Este trabalho gera *embeddings* contextualizados usando o $BERTimbau$ e os utiliza para calcular distâncias ou similaridade 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.

2.5. 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 na literatura. Os trabalhos são ordenados cronologicamente nas linhas da tabela. Eles são comparados conforme os aspectos das soluções de PLN que consideramos mais relevantes, o quais aparecem nas colunas. A primeira coluna **Autor e Ano** define o nome dos autores e ano de publicação do trabalho avaliado. A segunda coluna **Área de aplicação** descreve o domínio dos documentos analisados. A terceira coluna **Objetivos** refere-se à proposta de cada trabalho. O tipo de ***Embedding*** utilizado por cada trabalho é indicado na quarta coluna. A quinta coluna **Abordagem** refere-se aos métodos utilizados na proposta. Por fim, a coluna **Ferramentas** cita as ferramentas utilizadas no desenvolvimento da proposta.

Tabela 1. Tabela comparativa

Autor e Ano	Área de aplicação	Objetivos	*Embeddings*	Abordagens	Ferramentas
[Benício 2020]	Médica / Obstetrícia	Detectar termos clínicos	N/A	*Stemming*, *Levenshtein*	DeCS
[Bertalan e Ruiz 2020]	Juridica	Predizer resultados judiciais	GloVe	*Labeling*, TF-IDF, ML supervisionado	NLTK
[Sorato et al 2020]	Posts em Micro-blogs	Minerar padrões semânticos para analisar e classificar discursos	GloVe	*PoS-Tagging*, TF-IDF	spaCy, Scikitlearn, NLTK
[Santo 2021]	Médica	Sugestão de diagnóstico	N/A	*Label Encoding*, *One Hot Encoding*, ML supervisionado	NLTK, spaCy, DeCS
[Martins 2021]	Financeira	Detectar identificadores financeiros	N/A	OCR, *Stemming*	NLTK
[Braz-Junior and Fileto et al 2021]	Perguntas de QA	Analisar coerência de discursos	BERT	*POS-Tagging*	*nlpnet*, *scipy*, Google Colab
Nosso trabalho	**Médica**	**Minerar Padrões morfo-semânticos**	**BERT**	**ML não supervisionado**	**spaCy, DBpedia, Google Colab**

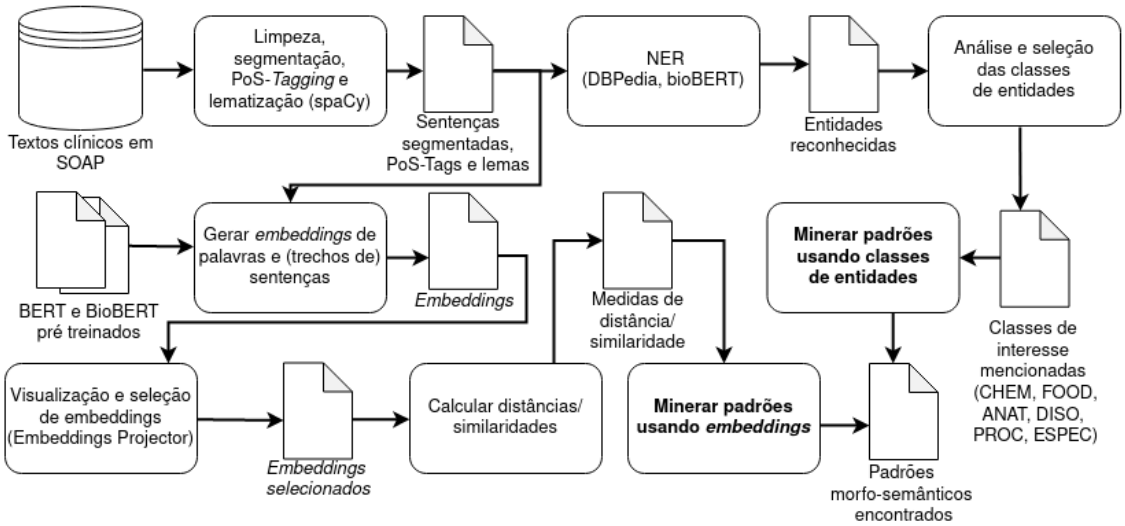
Nenhum dos trabalhos analisados usa *embeddings* contextualizados. [Bertalan and Ruiz 2020] e [Sorato et al. 2020] utilizam *embeddings* estáticos como GloVe para, respectivamente, predizer resultados judiciais e minerar padrões em textos, enquanto [Benício 2020], [Santos 2021] e [Martins 2021] não usam *embedding* nenhum.

Nossa proposta explora NER para identificar e classificar certas menções relevantes em textos (a medicamentos, comidas, estruturas anatômicas, especialidades e procedimentos médicos). Alternativamente, usa *embeddings* contextualizados produzidos pelo $BERTimbau$ (que capturam contexto textual em ambas as direções). As classes de NER e os *embeddings* permitem determinar casamento ou similaridade de sentidos, respectivamente, em métodos alternativos para minerar padrões morfo-semânticos em textos.

4. Processo e algoritmos para minerar padrões

Esta seção descreve as etapas do processo e algoritmos propostos neste trabalho para minerar padrões morfo-semânticos em textos clínicos. A Figura 4 ilustra o fluxo de informação proposto e avaliado neste trabalho para minerar os padrões em textos clínicos no padrão SOAP cedidos por uma empresa que presta serviços a operadoras de plano de saúde, embora o processo proposto possa ser aplicado a textos de outras fontes.

Figura 4. Processo para mineração de padrões morfo-semânticos em textos clínicos.



4.1. Limpeza, segmentação de sentenças, *PoS-Tagging* e lematização

A primeira etapa do processo proposto é o pre-processamento dos documentos (Textos Clínicos). Nesta etapa primeiro se faz a padronização dos textos, mediante remoção de certos caracteres especiais (e.g.,$\backslash n$), 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 documento, lista de seus tokens, lista de lemmas, lista de POS-*Tags*, lista de verbos e a lista de sentenças do documento. Esses dados são utilizadas para facilitar processo de geração dos *embedding* e reconhecimento de entidades nomeadas relevantes para mineração de padrões.

4.2. Reconhecimento de Entidades - NER

A tarefa de NER é realizada utilizando os dados gerados na etapa anterior (descrita na Subseção 4.1). As sentenças dos documentos são submetidas a duas soluções de NER para a identificação das entidades clínicas. A primeira é o $BioBERTpt$, o qual consiste de um modelo do BERT com ajuste fino para efetuar NER em textos do domínio clínico na língua portuguesa. O uso desta ferramenta de NER foi feito seguindo o exemplo postado pelo autor em seu repositório no github [1]. A segunda solução utilizada foi consultas pelas

[1]Disponível em `https://github.com/HAILab-PUCPR/BioBERTpt`

palavras encontradas no textos clínicos na DBPedia através de seu *endpoint* SPARQL.

A Figura 5 apresenta um exemplo de consulta desenvolvida em nosso trabalho buscando recursos rotulados pela palavra "Cetoprofeno". Como era desejado que fossem reconhecidos apenas recursos de ontologias registradas em português, criamos uma consulta que filtra os resultados retornados que contenham estas características. O filtro para ontologias verifica se o tipo retornado pela DBpedia contem a *string* "*http://dbpedia.org/ontology/*". Já o parâmetro "*@pt*" verifica se a palavra passada na consulta tem correspondência em português. Ainda assim, é possível que a palavra corresponda com múltiplas entidades de áreas diferentes, retornando uma lista extensa de resultados de diferentes áreas de conhecimento.

Figura 5. Consulta SPARQL feita na DBpedia para a palavra 'Cetoprofeno'

```
select distinct ?type
where {
     ?i rdfs:label "Cetoprofeno"@pt ;
     a ?type .
     FILTER (strstarts(str(?type),str("http://dbpedia.org/ontology/")))
     FILTER (strstarts(str(?i), str("http://dbpedia.org/resource/Ketoprofen")))
}
```

Após o término desses procedimentos, os dados obtidos são armazenados em uma estrutura contendo para cada palavra consultada o resultado obtido pelo NER. Esses resultados são utilizados para realizar a classificação das entidades reconhecidas.

4.3. Análise e seleção das classes de entidades

Nesta etapa, verificamos os resultados obtidos pelas 2 ferramentas de NER utilizadas na Subseção 4.2. O objetivo desta verificação é selecionar a ferramenta que reconhece corretamente a maior quantidade de entidades mencionadas nos textos clínicos. Após a seleção da ferramenta, as entidades reconhecidas por ela são separadas em classes de interesse utilizando a classificação proposta por [Campillos-Llanos et al. 2021]. Em nosso trabalho foram adicionados outras 2 classes de interesse: especialidades médicas (SPEC) e alimentos (FOOD). Estas 2 classes foram adicionadas pois verificamos que em nossos documentos os profissionais realizam orientações de ingestão de alimentos e indicações para outras especialidades médicas. Logo, poderíamos minerar padrões usando essas 2 classes de interesse adicionadas. Após a classificação das entidades nomeadas mencionadas, a lista com a respectiva classe de cada palavra dos documentos é concatenada na estrutura semi-estruturada criada na Subseção 4.1. Esta classificação é utilizada no processo de mineração de padrões usando classes de interesse.

4.4. Gerar *embeddings* de palavras e (trechos de) sentenças

Nesta etapa, as sentenças dos documentos 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 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 concatenação dos *embeddings* de seus *tokens*. Janelas são fragmentos de sentenças centrados em uma classe de interesse, 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 tomados como a média (*MEAN pooling*) das 4 últimas camadas do BERT, gerando embeddings com 4.096 valores (dimensões).

Algoritmo 1: Criar lista de janelas

Data: Lista de sentenças("$idSentenca$","$tokens$", "$postagging$", "$entidadeReconhecida$", "$classeDeInteresse$", "$posER$") 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.      posER ← reg["posER"]
6.      entidadeReconhecida ← reg["entidadeReconhecida"]
7.      classeDeInteresse ← reg["classeDeInteresse"]
8.      texto ← juncao(tokens)
9.      embSentenca ← getEmbeddingsText(texto)
10.     tamJanela ← 1
11.     janelaInf, janelaSup, expande ←
          expandeJanela(token, posER, posER)
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, entidadeReconhecida, classeDeInteresse]
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 classe de interesse. As sentenças relevantes (SR) utilizadas nessa tarefa são todas as que possuam uma palavra classificada como descrito na Subseção 4.3. Uma lista de registros é criada para armazenar as janelas com seus *embeddings*. Todas as sentenças são percorridas para gerar os *embeddings*. O registro de cada sentença da lista é composto

pelo id da sentença, lista de *tokens*, posição da entidade reconhecida, a própria palavra reconhecida e classe de interesse da entidade reconhecida. 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 e a classe de interesse 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.

4.5. Visualização e seleção de *embeddings*

Nesta etapa, os dados gerados como explicado na seção 4.4 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., classe morfossintática ou classe de sentido de *embedding* de palavra, classe de interesse de um *embedding* de palavra, classes de interesse presentes na sentença).

O *Embedding Projector* possibilita validar os *embeddings* carregados. Isso pode ser feito usando uma das técnicas apresentadas na Subseção 2.5 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. Se o conjunto de dados apresentar alguma propriedade relevante ou desejável, o conjunto de dados validado é separado para ser utilizado na tarefa na mineração de padrões.

4.6. Calcular distâncias/similaridades

O cálculo das medidas de similaridade e distância é realizado para cada documento. São calculadas a distância (Euclidiana e Manhathan) e a similaridade (cosseno) para todos os pares de *embeddings* de palavras. Todas as distâncias entre pares de *embeddings* ficam pré-calculadas e armazenadas para reuso pela próxima e última etapa do processo proposta, 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 entidade reconhecida pertencente a uma classe de interesse. Como entrada é utilizada a lista criada no Algoritmo 1 contendo os *embeddings* de janelas formado por id da sentença ("idSentenca"), sentença, janela, tamanho da janela ("$compJanela$"), lista de *embeddings* ("$embJanela$") e classe de interesse. Todos os elementos da lista de *embeddings* de janelas são percorridos e calculadas as medidas de distâncias/similaridade dos *embeddings* da janelas entre si ($n^2 - n$ comparações). 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, classe de interesse, id da sentença comparada, sentença comparada, tamanho da janela comparada, entidade reconhecida 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 ("$idSentenca$", "$sentenca$", "$janela$", "$compJanela$", "$embJanela$", "$entidadeReconhecida$", "$classeDeInteresse$") 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.      entidadeReconhecida_a ← reg_a["entidadeReconhecida"]
9.      classeDeInteresse_a ← reg_a["classeDeInteresse"]
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.         entidadeReconhecida_b ← reg_b["entidadeReconhecida"]
17.         classeDeInteresse_b ← reg_b["classeDeInteresse"]
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,
                  entidadeReconhecida_a, classeDeInteresse_a,
                  idSentenca_b, sentenca_b, janela_b, compJanela_b,
                  entidadeReconhecida_b, classeDeInteresse_b, medida]
21.             regcomp_b ← [idSentenca_b, sentenca_b, janela_b, compJanela_b,
                  entidadeReconhecida_b, classeDeInteresse_b,
                  idSentenca_a, sentenca_a, janela_a, compJanela_a,
                  entidadeReconhecida_a, classeDeInteresse_a, medida]
22.             listaRegistro.append(regcomp_a)
23.             listaRegistro.append(regcomp_b)
```

4.7. Mineração de padrões morfo-semânticos

Finalmente, a tarefa de minerar padrões morfo-semânticos pode ser realizada em torno de *embeddings* de palavras ou *embeddings* de entidades reconhecidas pertencentes as classes de interesse nas sentenças dos textos clínicos. A tarefa 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 4.6).

O Algoritmo 3 apresenta na forma de pseudocódigo o programa criado para a tarefa de mineração de padrões usando medidas de distâncias/similaridade de *embeddings* de palavras de sentenças que contenham uma determinada classe de interesse. Como entrada é utilizado a lista criada no Algoritmo 2 contendo pares de janelas ($janela_1, janela_2$) comparadas e sua medida. Um dicionário vazio é criado para armazenar as ocorrências de janelas e todas as medidas entre janelas são percorridas. Para medida entre janelas delimitado pelo valor do *threshold* é gerado um identificador formado pela concatenação do id da sentença, a classe de interesse e a janela. Este identificador é a chave para a entrada no dicionário de ocorrências. Se o identificador não existir no dicionário, uma entrada para ele é criada com um conjunto vazio. Para cada ocorrência de um identificador é adicionado o par de janela avaliado ao conjunto de seu respectivo identificador. Por fim a contagem de pares de janelas de cada item do dicionário é realizada para identificar qual janela apresenta a maior ocorrência.

Algoritmo 3: Mineração de Padrões

Data: Lista de medidas de entre janelas("$idSentenca_1$", "$sentenca_1$", "$janela_1$", "$compJanela_1$", "$entidadeReconhecida_1$", "$classeDeInteresse_1$", "$idSentenca_2$", "$sentenca_2$", "$janela_2$", "$compJanela_2$", "$entidadeReconhecida_2$", "$classeDeInteresse_2$", "$medida$")

Result: dicionário de ocorrências de padrões semânticos em janelas

```
1.  dicionario ← dict()
2.  foreach index, reg in listaJanelas do
3.      idSentenca ← reg["idSentenca_1"]
4.      entidadeReconhecida ← reg["entidadeReconhecida_1"]
5.      janela ← reg["janela_1"]
6.      medida ← reg["medida"]
7.      if medida ≥ threshold then
8.          identificador =
              idSentenca + "__" + entidadeReconhecida + "__" + 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 realizados para avaliar a nossa proposta de mineração de padrões morfo-semânticos. Esses experimentos visam verificar a existência e as características de padrões morfo-semânticos nos textos clínicos escritos por profissionais da

área da saúde ao realizar atendimentos.

5.1. Implementação

A proposta foi implementada na linguagem de programação Python versão 3.7.13, sobre notebooks do ambiente de execução Google colaboratory[2]. 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[3], 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 $BERTimbau$[4] [Souza et al. 2020] no tamanho *large*, no formato "*cased*" (com caracteres maiúsculos e minúsculos) disponíveis gratuitamente. Para a tarefa de segmentação de sentenças e POS-*Tagging* das palavras utilizamos a ferramenta de PLN spaCy[5] versão 3.2.0.

5.2. Conjunto de dados

Os dados de atendimentos médicos disponibilizadas para realização deste trabalho foram fornecidos por uma empresa que presta serviços a operadoras de plano de saúde, com a supervisão de um dos sócios de tal empresa. Os dados foram recebidos em arquivos no formato CSV, contendo apenas os campos de texto livre dos prontuários. Desta forma nomes, documentos e outras informações sensíveis dos pacientes não foram compartilhadas.

Após realizar a extração e a limpeza dos dados, identificamos que o campo Plano (P) possuía textos mais estruturados e em maior quantidade. Desta forma escolhemos usar apenas este campo para a realização do trabalho. Ao extrair o campo Plano do arquivo CSV, foram obtidos 458 documentos. Foi observado que existiam muitos documentos idênticos repetidos nos textos extraídos do campo Plano (89 no total), mostrando que os profissionais reaproveitam os textos em casos de diagnósticos semelhantes. Como documentos idênticos não trazem relevância para a mineração de padrões e também geram uso desnecessário de disco e tempo de processamento, estes documentos foram removidos dos textos clínicos. Após a remoção dos documentos idênticos, a quantidade de documentos ficou em 369. A Tabela 2 apresenta as principais estatísticas encontradas no campo Plano dos textos clínicos depois da remoção dos documentos idênticos.

Tabela 2. Estatísticas do conjunto de dados

	Sentenças	**Palavras**	**Tokens BERT**	**Palavras desconhecidas**
Quantidade Total	1.054	12.912	19.710	3.722
Média por Doc.	3,81	46,61	71,16	13,44
Desvio padrão por Doc.	2,15	36,89	49,73	8,22

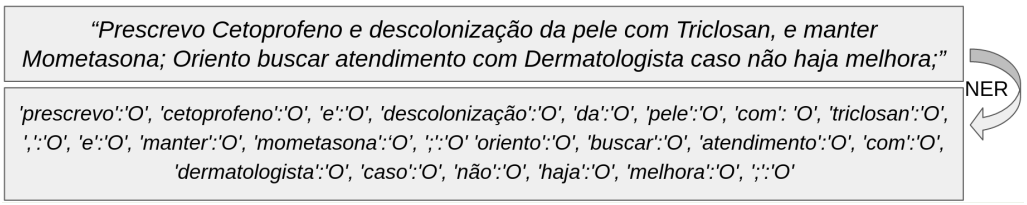
[2]https://colab.research.google.com/notebooks/intro.ipynb
[3]https://huggingface.co/transformers/index.html
[4]https://github.com/neuralmind-ai/portuguese-bert/
[5]https://spacy.io/

5.3. Reconhecimento de Entidades Nomeadas

Após a realização dos testes preliminares, iniciamos os testes com as ferramentas de NER. A primeira ferramenta utilizada e que estava sendo, até o momento, considerada como a principal ferramenta de NER da proposta foi o $BioBERTpt$. Escolhemos utilizar a versão *all* do modelo treinado. Ao usar o exemplo do autor, tivemos que fazer algumas alterações pois o exemplo faz uso de arquivos que não constam no diretório da biblioteca HuggingFace.

Com o exemplo funcionando, iniciamos a aplicação em nossos textos clínicos. Contudo, após testar 3 diferentes maneiras de implementação para usar o modelo, a solução não foi capaz de identificar nenhuma entidade nos nossos textos clínicos. Por esse motivo o $BioBERTpt$ não foi usado como solução de NER e não foram feitas métricas com essa ferramenta. Um exemplo com um documento sendo passado isoladamente pelo modelo para um teste simples é ilustrado na Figura 6, onde a tag 'O' inserida após cada palavra significa que esta palavra está fora do vocabulário *"Out of Vocabulary"* do modelo.

Figura 6. Exemplo de documento retirado dos Textos Clínicos



Para suprir a falta de entidades reconhecidas, recorremos ao uso de uma ferramenta de propósito geral: a DBpedia. Para usar sua ferramenta *web*, criamos uma rotina que faz consultas SPARQL ao seu *endpoint*, usando a biblioteca SPARQLWrapper do Python. Essas consultas foram feitas apenas para palavras alfanuméricas com tamanho maior que 2, para evitar fazer consultas com os artigos das sentenças. Usamos o padrão alfanumérico para fazer consultas com palavras como "COVID19" e não fazer consultas com caracteres especiais. Após o uso desse padrão e do limite de tamanho aplicado ficamos com um total de 1.001 palavras, sem repetições, para serem consultadas na DBpedia.

Para selecionar nos resultados das consultas realizadas as entidades da área clínica, foram anotados manualmente os tipos retornados para palavras da área clínica (e.g. remédios, doenças, procedimentos, etc...) mencionadas nos nossos textos clínicos. Em seguida, os tipos anotados foram divididos dentro das classes de interesse mencionadas na Subseção 4.3. A Tabela 3 apresenta as estatísticas das entidades reconhecidas usando consultas SPARQL à DBpedia, por classe de interesse.

Tabela 3. Estatísticas das entidades reconhecidas pela DBpedia em relação ao conjunto total de palavras (1.001).

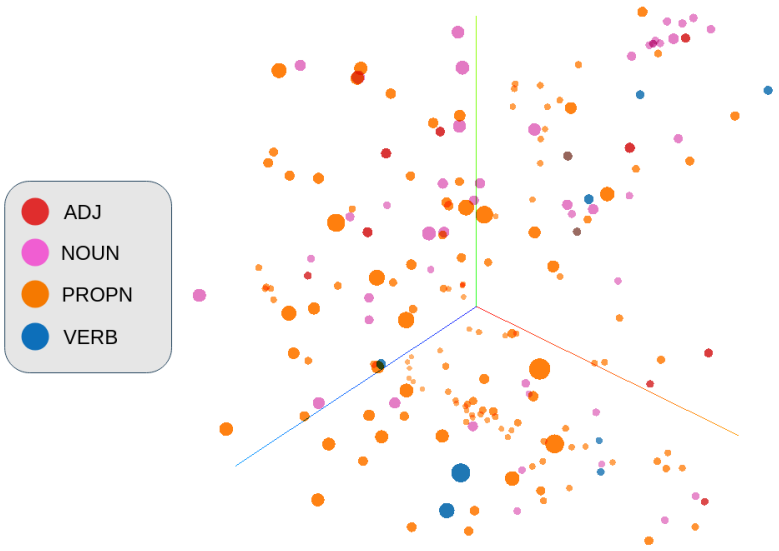
Classes	CHEM	DISO	ANAT	PROC	FOOD	Total
#Entidades reconhecidas	40	22	3	8	2	76
%Entidades reconhecidas por palavras consultadas	4,00%	2,20%	0,30%	0,80%	0,20%	7,59%

5.4. Visualização e mineração de padrões

Foi criado um experimento para analisar as classes morfo-sintáticas dos *embeddings* próximos às entidades reconhecidas. Nesse experimento criamos uma visualização no *Embeddings Projector* utilizando os *embeddings* de 10.690 palavras, gerados pela concatenação das 4 últimas camadas do $BERTimbau$ (4.096 parâmetros por palavra). Usamos os 2 algoritmos de clusterização da ferramenta e o que apresentou um melhor resultado de agrupamento com relação às classes de interesse foi o UMAP. Para analisar as palavras próximas às classes de interesse, fizemos um recorte nos dados em um agrupamento de entidades reconhecidas classificadas como CHEM.

Após o recorte nos dados obtivemos uma visualização com 202 pontos. Nesta visualização em três dimensões utilizando o algoritmo PCA configuramos a cor dos pontos para as suas respectivas PoS-*taggings*, com o intuito de observar possíveis verbos. A Figura 7 ilustra a visualização obtida com a respectiva legenda de cores. Nela é possível observar que a maioria dos pontos foi classificada pelo spaCy como substantivos (NOUN) e nomes próprios (PROPN). Logo, as entidades reconhecidas classificadas como medicamentos receberam *embeddings* contextualizados com valores próximos aos *embeddings* de palavras com semântica semelhante. Também é possível observar que existem poucos pontos (5) representando verbos. Ao fazer a verificação destes verbos, notamos que o spaCy classificou incorretamente os seguintes medicamentos como verbos: allegra, Dipirona, Loratadina e Azitromicina. Isso mostra que a tarefa de PoS-*tagging* realizada cometeu erros de classificação, mas os *embeddings* contextualizados desses medicamentos também receberam valores próximos aos *embeddings* de palavras semanticamente semelhantes.

Figura 7. Recorte feito na área de interesse da visualização dos *embeddings* de 10.690 palavras coloridas pela classe morfo-sintática



Para verificar os padrões contidos nos textos clínicos, usamos o Algoritmo 3 para obter as palavras mais similares às entidades reconhecidas mencionadas. Filtramos as comparações realizadas utilizando o *threshold* de $0,8$, usado por [Sorato et al. 2020]. A

Tabela 4 mostra as quantidades de entidades comparadas, separadas por classes de interesse, assim como as médias das medidas de distâncias e similaridade obtidas.

Tabela 4. Quantidades e médias das distâncias e similaridade de entidades nomeadas com *threshold* maior ou igual $0,8$.

Classe	Qtd.	Média cos	Média Euc	Média Man
CHEM	896	0.828	10.8013	267.8409
DISO	14	0.8068	12.5976	315.3274
PROC	106	0.9846	3.7027	94.8284

Na Tabela 5 podemos visualizar as entidades reconhecidas mencionadas nos textos clínicos com a maior quantidade de vezes que ela teve a similaridade dos seus *embeddings* acima do threshold utilizado. A primeira entidade da lista (Cetoprofeno) tem 132 ocorrências de palavras similares a ela. A Tabela 6 lista essas 132 ocorrências agrupadas por palavras iguais.

Tabela 5. Lista das 10 palavras/entidades com maior quantidade de ocorrências de similaridades dos *embeddings*, classes, sentenças e quantidades.

Entidade Reconhecida	Classe	Sentença	Qtd.
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Zyrtec , Avamys , Bromexina , Lavagem nasal com SF) ;	132
Paracetamol	CHEM	Dip 1 g 6/6h , SN + **Paracetamol** 750	76
ainda	PROC	Oriento **ainda** a retornar o contato em caso de piora dos sintomas ;	50
Cetoprofeno	CHEM	Conduta : Prescrevo **Cetoprofeno** e descolonização da pele com Triclosan , e manter Mometasona ;	48
Desloratadina	CHEM	**Desloratadina** 0,5 mg ml 10 ml dia , se tosse seca	42
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Baclofeno , Tylex) ;	40
Cetoprofeno	CHEM	Prescrevo sintomáticos SN (Dipirona , **Cetoprofeno** , Zyrtec) ;	40
ibuprofeno	CHEM	Substituo **ibuprofeno** por 3 dias de cetoprofeno 100 mg 12/12h associado a dipirona em dose analgésica (1 g) .	38
cetoprofeno	CHEM	Substituo ibuprofeno por 3 dias de **cetoprofeno** 100 mg 12/12h associado a dipirona em dose analgésica (1 g) .	38
Mometasona	CHEM	Conduta : Prescrevo Cetoprofeno e descolonização da pele com Triclosan , e manter **Mometasona** ;	36

Tabela 6. Lista das 132 palavras com *embeddings* similares à Entidade Reconhecida 'Cetoprofeno'

Palavra	Qtd.
Dipirona	36
Cetoprofeno	30
Paracetamol	18
Bromexina	18
cetoprofeno	12
Baclofeno	6
ibuprofeno	6
Ciprofloxacino	6

5.5. Discussão

Os resultados ruins dos experimentos de NER com o modelo $BioBERTpt$ (em termos de cobertura e precisão) podem se devidos ao fato do software estar mal documentado, atrapalhando o uso correto de seus modelos.

Dessa forma não conseguimos reconhecer corretamente todas as entidades mencionadas mesmo para um conjunto de dados pequeno. O conjunto de dados disponibilizado, além de ser pequeno, continha muito reuso de sentenças e muitos caracteres especiais, diminuindo a eficácia do PLN. Também não tivemos acesso a um corpus padrão ouro para realizar uma métrica de acurácia das ferramentas de NER.

A tarefa de visualizar e selecionar *embeddings* (descrita na Subseção 4.5) possibilitou identificar alguns agrupamentos. Principalmente as visualizações de *embeddings* de algumas classes de interesse mostraram agrupamentos. 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 *embeddings* da concatenação das 4 últimas camadas do $BERTimbau$, de acordo com a recomendação de [Devlin et al. 2019], que gera *embeddings* com 4.096 valores (dimensões). Essa escolha fez com que ocasionalmente as estruturas criadas nos experimentos consumissem todos os 12 GB de memória disponibilizados. 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 clínicos visando dar suporte a classificação não supervisionada de tais textos.

Realizamos a mineração de padrões usando medidas de distância e similaridade entre os *embeddings* de palavras, assim como usando casamento de classes de interesse das entidades reconhecidas nos textos.

O estudo e compreensão dos métodos, técnicas e ferramentas de PLN permitiu desenvolver algoritmos para mineração de padrões morfo-semânticos em textos clínicos. Os resultados dos algoritmos demostram que o BERT usa o contexto dos documentos para gerar os *embeddings* dos medicamentos próximos aos *embeddings* de outras palavras mencionadas nos mesmos contextos textuais, tais como doenças tratadas com os respectivos medicamentos. Isso não permite discriminar medicamentos e doenças, por exemplo, em grupos distintos de *embeddings*. Também observamos que os *embeddings* gerados para os verbos geralmente usados pelos profissionais da saúde para fazer prescrições aos pacientes não são necessariamente próximos dos *embeddings* dos medicamentos prescritos.

Os algoritmos de mineração de padrões morfo-semânticos em textos clínicos precisam ser reavaliados utilizando uma base de dados padrão ouro. Um padrão ouro possibilitaria uma análise quantitativa dos resultados, pois sem ele apenas a análise qualitativa dos padrões foi realizada.

Trabalhos futuros incluem: (i) aprimoramento dos algoritmos para mineração de padrões morfo-semânticos baseados em proximidade de *embeddings* e classes de interesse obtidas via NER; (ii) comparação mais exaustiva desses algoritmos, usando bases de dados maiores e mais diversificadas; (iii) desenvolvimento e uso de uma ferramentas de NER específicas para textos clínicos; (iv) utilizar modelos de *word embeddings* mais recentes como *BLOOM*[6] e (v) investigação das possibilidades de aplicação da abordagem proposta a textos de outros domínios, tais como textos jurídicos, científicos e de literatura.

Referências

Benício, D. H. P. (2020). Aplicação de mineração de texto e processamento de linguagem natural em prontuários eletrônicos de pacientes para extração e transformação de texto em dado estruturado. Master's thesis, Universidade Federal do Rio Grande do Norte.

Bertalan, V. G. F. and Ruiz, E. E. S. (2020). Predicting judicial outcomes in the brazilian legal system using textual features. In *DHandNLP@ PROPOR*, pages 22–32.

Campillos-Llanos, L., Valverde-Mateos, A., Capllonch-Carrión, A., and Moreno-Sandoval, A. (2021). A clinical trials corpus annotated with umls entities to enhance the access to evidence-based medicine. *BMC medical informatics and decision making*, 21(1):1–19.

Cordeiro, B. C. (2019). *BERT e Word2Vec: Uma análise inferencial e computacional na classificação de textos com redes neurais convolucionais*. PhD thesis, Universidade Federal do Rio de Janeiro.

de Abreu, J. C. O., Fileto, R., Ngomo, A.-C. N., Röder, M., Wittwer, M., and Saggion, H. (2017). Characterizing mention mismatching problems for improving recognition results. In *Proceedings of the 19th International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '17, page 85–94, New York, NY, USA. Association for Computing Machinery.

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

de Souza, A. D. and Felipe, E. R. (2021). Processamento de linguagem natural aplicado à anamneses do domínio da ginecologia. -.

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.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195.

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*.

Li, J., Sun, A., Han, J., and Li, C. (2022). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(01):50–70.

Lopes, A. A. (2020). Prontuário orientado por problemas e evidências (pope). -.

Ly, A., Uthayasooriyar, B., and Wang, T. (2020). A survey on natural language processing (nlp) and applications in insurance. *arXiv preprint arXiv:2010.00462*.

Martins, L. J. (2021). Cálculo de indicadores financeiros com auxílio do processamento de linguagem natural. -.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

Oliveira, I. L., Fileto, R., Speck, R., Garcia, L. P., Moussallem, D., and Lehmann, J. (2021). Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624.

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

Pinto, V. B. and Sales, O. M. M. (2017). Proposta de aplicabilidade da preservação digital ao prontuário eletrônico do paciente. *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, 15(2):489–507.

Piris, Y. and Gay, A.-C. (2021). Customer satisfaction and natural language processing. *Journal of Business Research*, 124:264–271.

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.

Ridgway, J. P., Uvin, A., Schmitt, J., Oliwa, T., Almirol, E., Devlin, S., and Schneider, J. (2021). Natural language processing of clinical notes to identify mental illness and substance use among people living with hiv: retrospective cohort study. *JMIR Medical Informatics*, 9(3):e23456.

Santos, Y. J. A. d. (2021). Avaliação de técnicas de aprendizado supervisionado no auxílio a diagnósticos médicos. -.

Sevgili, Ö., Shelmanov, A., Arkhipov, M., Panchenko, A., and Biemann, C. (2022). Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 1(Preprint):1–44.

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.

Sorin, V., Barash, Y., Konen, E., and Klang, E. (2020). Deep learning for natural language processing in radiology—fundamentals and a systematic review. *Journal of the American College of Radiology*, 17(5):639–648.

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.

Speck, R. and Ngonga Ngomo, A.-C. (2014). Ensemble learning for named entity recognition. In Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., and Goble, C., editors, *The Semantic Web – ISWC 2014*, pages 519–534, Cham. Springer International Publishing.

Wang, J., Deng, H., Liu, B., Hu, A., Liang, J., Fan, L., Zheng, X., Wang, T., Lei, J., et al. (2020). Systematic evaluation of research progress on natural language processing in medicine over the past 20 years: Bibliometric study on pubmed. *Journal of medical internet research*, 22(1):e16816.

Weed, L. L. (1968). Medical records that guide and teach. *New England Journal of Medicine*, 278(12):652–657.

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*, abs/2005.06249(1).