Capítulo 1

Processamento de Linguagem Natural

Helena Caseli, Cláudia Freitas e Roberta Viola

Abstract

Natural language processing (NLP) is an interdisciplinary research area, mainly involving Computing and Linguistics, which aims to process verbal, written or spoken language produced by human beings. This chapter focus on processing texts written in Brazilian Portuguese, with the presentation of toolkits and specific resources for that language. Here we will cover pre-processing steps, corpus annotation and generation of representations and language models most used today in important applications present in our daily lives such as conversational agents (chatbots), virtual assistants and semantic analyzers capable of returning, for example, the polarity (valence) of a post on a social network.

Resumo

O processamento de linguagem natural (PLN) é uma área de pesquisa interdisciplinar, envolvendo principalmente a Computação e a Linguística, que visa processar a linguagem verbal, escrita ou falada, produzida pelos seres humanos. Este capítulo foca no processamento de textos escritos em português do Brasil, com a apresentação de toolkits e recursos específicos para esse idioma. Aqui serão abordadas etapas de pré-processamento, anotação de corpus e geração de representações e modelos de linguagem mais utilizados na atualidade em importantes aplicações presentes em nosso dia-a-dia como os agentes conversacionais (chatbots), os assistentes virtuais e os analisadores semânticos capazes de retornar, por exemplo, a polaridade (valência) de uma postagem em uma rede social.

1.1. Introdução

Este minicurso tem como propósito apresentar uma introdução ao Processamento Automático das Línguas Naturais (PLN). O PLN pode ser situado como uma subárea da computação e da linguística (McShane e Nirenburg 2021), onde o objetivo principal é desenvolver modelos computacionais e recursos linguísticos úteis para a automatização do processamento das línguas humanas. Esses modelos podem ser produzidos como uma "atividade fim" para explicar um fenômeno linguístico, ou como uma "atividade meio",

onde visa-se um produto ao final. Neste curso, o foco está no processamento de textos escritos em português do Brasil, utilizando recursos disponíveis livremente, com vistas a gerar um produto de PLN.

No PLN, palavras interessam enquanto unidades portadoras de *sentido*. No entanto, o sentido das palavras é instável, isto é, pode variar ao longo do tempo, do espaço, de grupos de falantes, de diferentes áreas do conhecimento. Por isso, na prática, é limitada a busca por uma representação única e verdadeira do sentido (veja-se por exemplo (Kilgarriff 1997)).

A análise linguística não é linear, embora uma apresentação linear (dos morfemas às palavras, das palavras às frases, das frases aos textos; ou da morfologia à sintaxe, da sintaxe à semântica etc) funcione bem em termos didáticos. A frase abaixo[1] é um bom exemplo desta não linearidade:

> Com estas palavras, André Coruja, [...] além de *quebrar o gelo* **que** havia esfriado o clima, devolveu ao recinto a eloquência necessária para que a sessão continuasse [...]

Para compreender o sentido da frase, precisamos que *quebrar o gelo* seja entendido como uma unidade de sentido (algo próximo a *descontrair*), mas também precisamos que *quebrar o gelo* seja entendido como três palavras distintas, pois apenas o *gelo* havia esfriado o clima. Ou seja, para a semântica, precisamos que *quebrar o gelo* seja uma unidade indecomponível, mas para a sintaxe precisamos da decomposição, pois precisamos de *gelo* com o seu sentido literal.

Semântica e sintaxe são dois dos níveis de processamento das línguas naturais originalmente definidos para operacionalizar o entendimento e a automatização. Contudo, é curioso perceber como essas dimensões da língua se articulam de maneira orgânica no PLN feito nos últimos anos: os vetores de palavras contextuais[2] (*contextual embeddings*) materializam a instabilidade e multiplicidade de sentidos, e o processamento multicamadas, profundo, com a não-linearidade.

Metalinguagens linguísticas (classes como *verbo*, *sujeito*, *agente*) são produtos humanos, situados historicamente e motivados por problemas específicos (que nunca foram o processamento automático de uma língua). Esse aspecto pode funcionar como um estímulo para experimentações relativas à classificação linguística.

Línguas humanas são sistemas complexos, abertos e dinâmicos. De fato, uma das características das línguas que as fazem serem robustas não é a sua precisão, mas, pelo contrário, a sua vagueza. E por vagueza entendemos aqui a existência de limites difusos, pouco precisos, sobretudo no que se refere ao sentido. Um rápido exercício: que palavra nos é mais "útil": *coisa* ou *neurônio*? *Tomar* (tomar um banho, um suco, um porre, uma surra, vergonha, juízo, as dores ...) ou *esverdear*? Línguas humanas sofrem interferências diversas, havendo sempre uma tensão entre o seu caráter regular e

[1] O trecho foi retirado de <http://www.overmundo.com.br/overblog/rock-paraense-no-diva>

[2] Também conhecidos como modelos de linguagem contextualizados (*contextualized language models*) dos quais BERT (Devlin et al. 2019) é um dos representantes mais populares.

previsível (em *desfazer*, *deslegitimar* e *desleal*, temos o elemento *des-* como um indicativo de reversão ou negação, e essa regularidade nos permite produzir e compreender uma palavra que nunca ouvimos antes, como *desestender* em "Preciso desestender a roupa do varal") e seu caráter irregular e idiossincrático (uma pizza *desgostosa* não é uma pizza sem sabor ou ruim).

Regularidade e irregularidade são igualmente parte das línguas, de qualquer língua, como demonstrado empiricamente pela lei de Zipf, nomeada em homenagem ao linguista George Kingsley Zipf (1902–1950). O que esta lei demonstra é que os dados linguísticos (ou, as palavras em um texto) têm uma distribuição desigual: sempre temos poucos casos com muita frequência (regularidade) e muitos casos de frequência baixíssima (irregularidade).

Uma consequência dessa característica é que abordagens de PLN baseadas exclusivamente em regras, por serem definidas com base na regularidade, estão fadadas a deixar muita coisa de fora. Se a intenção é poder trabalhar com um conjunto de textos inéditos, a irregularidade precisa ser considerada e tratada da maneira adequada. Por outro lado, lembrar que há regularidade e previsibilidade ajuda se a ideia for trabalhar em um ambiente de linguagem controlada.

Este minicurso visa, portanto, trazer uma apresentação do PLN nesse novo cenário em que os modelos automáticos baseados em contextos, frequências e, principalmente, redes neurais estão avançando o estado-da-arte no PLN e tornando o processamento de grandes quantidades de textos uma tarefa factível, gerando modelos úteis para além das portas da academia.

Esta seção está organizada como segue. Na seção 1.1.1 são apresentados os níveis originais do PLN. Na seção 1.1.2 serão listadas as principais abordagens utilizadas na área (simbólica, estatística, neural). Na seção 1.1.3 são listados os recursos geralmente utilizados, como os *corpora* (conjuntos de textos), as *word embeddings* e diversos recursos lexicais (como a WordNet). A seção 1.2 descreve as etapas de pré-processamento de textos usualmente realizadas para converter uma sequência de caracteres em informações úteis para aplicações de PLN. A seção 1.3 traz os principais formalismos da atualidade para representação de dados textuais. O aprendizado supervisionado é o tema da seção 1.4, com atenção especial ao processo de preparação dos dados a serem usados nesse aprendizado. Por fim, a seção 1.5 descreve como modelos neurais pré-treinados para o português podem ser refinados para algumas aplicações de PLN.

1.1.1. Níveis do Processamento de Linguagem Natural

O propósito do PLN é tornar os computadores aptos a processar a língua natural, para uma discussão aprofundada veja (McShane e Nirenburg 2021). Por *processar*, aqui, entende-se uma gama de aptidões próprias dos seres humanos que queremos incorporar às máquinas: entender, gerar, extrair conhecimento útil, comunicar-se, entre outros.

A grande quantidade de dados produzida e disponibilizada diariamente, em língua natural, só reforça a relevância e a urgência por processar esses dados e transformá-los em conhecimento. Há uma estimativa de que cerca de 80-90% dos dados produzidos está na forma de dados não estruturados (como áudio, vídeo, foto, texto), sendo as produções

textuais representantes de grande parte desses dados.[3] A esse fato soma-se o de que, cada vez mais, os dados textuais disponíveis são os gerados pelo próprio usuário em postagens de redes sociais e produções textuais despretensiosas e livres de qualquer rigor linguístico, o que traz novos e maiores desafios do que os que existiam nos primórdios do PLN.

Nesses primórdios, o PLN foi definido para ser processado em níveis: (1) Fonético/fonológico, preocupado na especificação e estudo dos sons das línguas; (2) Morfológico, interessado na definição e processamento das unidades linguísticas; (3) Sintático, responsável pela investigação e criação das regras que ordenam as unidades linguísticas e definem como elas se relacionam; (4) Semântico, que visa o estudo dos significados e (5) Pragmático/Discursivo, preocupado com a influência do contexto (e intenção do falante) na linguagem.

Assim, convencionou-se que as pesquisas e seus produtos, no PLN, fossem associados aos níveis acima. Um Reconhecedor de Voz (*Automatic Speech Recognition*, ASR) por exemplo, que é um sistema capaz de converter sinais de voz em texto (palavras), ou sua contraparte que converte as palavras em sinais de voz por meio de um Sintetizador de Voz (também referenciado como *text-to-speech*), são considerados produtos do nível fonético/fonológico apesar de hoje serem gerados por modelos neurais de muitas camadas nas quais as divisões em níveis linguísticos inexistem.

Nesse contexto, determinar a qual nível uma pesquisa ou um produto de PLN pertence tem sido uma tarefa cada vez mais difícil. Na realidade, essa dificuldade sempre existiu, pois a língua não deve ser vista como algo a ser processado dentro de caixinhas bem-definidas e sem sobreposição. Ao se considerar o exemplo do *quebrar o gelo* apresentado anteriormente, ficou claro que a divisão entre morfologia, sintaxe e semântica é muito nebulosa e não determinística uma vez que não é uma tarefa fácil se determinar onde começa e termina uma unidade linguística (papel da morfologia), dado o contexto, (papel da sintaxe) a fim de se determinar o significado naquela ocorrência (papel da semântica).

1.1.2. Abordagens mais utilizadas no Processamento de Linguagem Natural

Outra caracterização bastante usual nas pesquisas no PLN é a de associá-las a uma das abordagens mais utilizadas: simbólica (linguística), estatística ou neural.

A abordagem simbólica foi a primeira a ser proposta, quando o PLN surgia juntamente com as primeiras implementações computacionais a partir de 1950. Naquela época, os especialistas tinham em mãos seus conhecimentos sobre o processamento da língua natural e utilizaram meios simbólicos de expressar explicitamente esse conhecimento em formatos que as máquinas pudessem processar. Nesse momento, surgiram os léxicos de palavras, as primeiras gramáticas processáveis por computador e algumas bases de conhecimento.

Com os avanços na disponibilização de dados textuais, em especial os primeiros *corpora*[4], dados começaram a surgir em abundância permitindo que a estatística fosse usada para o reconhecimento de padrões (baseado em frequência e probabilidade) que

[3]<https://mitsloan.mit.edu/ideas-made-to-matter/tapping-power-unstructured-data>

[4]*Corpora* é o plural de *corpus*. *Corpus* é o nome que se dá a uma coleção de textos.

deram origem aos primeiros modelos estatísticos. Diversos algoritmos de aprendizado de máquina (AM) foram empregados para a geração dos modelos estatísticos. Hoje, tais modelos são denominados de modelos de AM tradicionais ou *shallow*. É desta época a primeira versão do Google Tradutor[5], que foi estatístico de 2006 a 2016 até ser substituído pela versão neural atualmente em uso. De fato, a abordagem tradicional de AM reinou soberana entre os anos de 1980 e 2010 quando as redes neurais artificiais roubaram a cena em diversas aplicações computacionais e também no PLN.

A abordagem neural (*deep learning* ou aprendizado profundo) é hoje a mais usualmente aplicada para se alcançar resultados de estado-da-arte no PLN. Embora, em sentido estrito, a abordagem neural também seja estatística, convencionou-se tratá-las como abordagens distintas. Isso ocorre porque, mesmo também sendo utilizado para reconhecer os padrões recorrentes, o aprendizado profundo lida com várias camadas de unidades de processamento capazes de aprender a partir de grandes quantidades de dados sem que sejam explicitamente programadas. Esse fato faz com que, na prática, o aprendizado profundo seja considerado uma abordagem distinta da estatística. Em outras palavras, enquanto os algoritmos tradicionais de AM especificam como o aprendizado deve ocorrer, nas redes neurais esse aprendizado é realizado em camadas de unidades de processamento (neurônios artificiais) de modo que não é possível saber, ao certo, qual dado de entrada levou a qual padrão aprendido. Além da falta de interpretabilidade do modelo, o poder computacional e a quantidade de dados que os modelos neurais demandam para se alcançar resultados tão impressionantes são limitantes para diversas aplicações e instituições.

Assim, uma alternativa que tem sido empregada com bastante sucesso na atualidade é utilizar modelos neurais pré-treinados a partir de quantidades de dados e poder computacional não disponíveis a todos, e realizar um refinamento (*fine-tuning*) de tais modelos para tarefas específicas para as quais dados e poder computacional são considerados acessíveis.

1.1.3. Recursos mais comuns no Processamento de Linguagem Natural

No aprendizado de máquina, seja ele *shallow* ou *deep*, um recurso essencial são os dados usados no treinamento dos modelos. No caso do PLN, o principal tipo de dado utilizado é um *corpus*. Um *corpus* é uma coleção de textos, por exemplo, um conjunto de *tweets* ou uma coletânea de obras literárias.[6]

Um *corpus* pode ser: comparável, quando trata de uma coleção de textos que versam sobre um mesmo assunto; paralelo, quando trata-se de versão em uma língua e sua(s) tradução(ões) para outra(s) língua(s); ou alinhado, quando além de ser paralelo o *corpus* possui indicações de qual sentença ou palavra fonte (no texto original), por exemplo, é a tradução de qual palavra ou sentença alvo (no texto traduzido). Além disso, também pode-se tipificar o *corpus* em relação à quantidade de línguas que ele abrange como: monolíngue, bilíngue ou multilíngue. Um exemplo de *corpus* paralelo português-inglês e português-espanhol é a coletânea de artigos de divulgação científica da revista Pesquisa

[5]<https://translate.google.com.br/>

[6]Grafias possíveis para a palavra *corpus* são sua versão no latim, em itálico (cujo plural é *corpora*) ou sua versão acentuada córpus que serve tanto para o singular quanto para o plural, em português.

FAPESP[7] conhecido como Corpus FAPESP[8] (Aziz e Specia 2011). Outros *corpora* para o português também estão disponíveis em repositórios como o Metatext[9].

Outro recurso bastante útil no PLN na atualidade são as *word embeddings* ou vetores de palavras. As *embeddings* são representações vetoriais estáticas onde cada palavra é representada como um ponto em um espaço *n* dimensional. Por exemplo, *embeddings* de 300 dimensões usam 300 valores reais para representar uma dada palavra. Essas representações podem ser aprendidas a partir de *corpus* por meio de contagem de frequências – como o GloVe (Pennington et al. 2014) – ou modelos neurais – como o Word2Vec (Mikolov et al. 2013). Por meio dessas representações vetoriais é possível encontrar similaridades sintáticas e semânticas inferidas no aprendizado com base no contexto de ocorrência das palavras no *corpus* usado no treinamento. Algumas *word embeddings* podem ser encontradas no repositório do NILC[10] (Núcleo Interinstitucional de Linguística Computacional).

Um avanço em relação às representações estáticas que são as *word embeddings* são os modelos de linguagem contextualizados como o BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019) e GPT-3 (Brown et al. 2020). Nesse caso, diferente das *word embeddings* onde cada palavra só possui um vetor que a representa, cada ocorrência da palavra tem sua própria representação. Uma das principais vantagens dos modelos de linguagem contextualizados é que palavras polissêmicas como *banco* podem ter seus diferentes significados preservados, enquanto nos vetores estáticos essa polissemia se perde quando apenas uma representação (a mais frequente) é preferida pelo modelo. Um dos modelos de linguagem mais famosos para o português do Brasil é o BERTimbau (Souza et al. 2020) disponível no repositório do HuggingFace[11].

Além dos *corpora* e dos recursos derivados deles por meio de aprendizado de máquina – *word embeddings* e modelos de linguagem contextualizados – há uma ampla gama de recursos linguístico-computacionais simbólicos e muito ricos, como as wordnets.

As wordnets são, como o nome indica, redes (*nets*) de palavras (*words*). Wordnets são consideradas bases de dados *lexicais*, porque contêm informações relacionadas ao léxico de uma língua[12]. Diferentemente de dicionários, que também lidam com palavras, wordnets não estão organizadas por palavras, mas por *synsets*. Synsets são conjuntos (*sets*) de sinônimos (*synonyms*) que se relacionam entre si por meio de relações semânticas como hiperonímia e hiponímia (relações hierárquicas), meronímia (parte – todo), dentre outras. As palavras *berrar* e *gritar*, por exemplo, integram um mesmo *synset*. Este *synset*, por sua vez, não é representado por uma outra palavra ou conceito, mas por um código. Por exemplo:

```
00912473-v {berrar, gritar}
```

O *synset* acima, por sua vez, é hipônimo – isto é, está um nível abaixo – do *synset*

[7]<https://revistapesquisa.fapesp.br/>

[8]<http://www.nilc.icmc.usp.br/nilc/tools/Fapesp%20Corpora.htm>

[9]<https://metatext.io/datasets-list/portuguese-language>

[10]<http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>

[11]<https://huggingface.co/neuralmind/bert-base-portuguese-cased>

[12]O léxico é a parte da língua que diz respeito às palavras. De forma simplificada, podemos entender uma língua como a combinação de um léxico (as palavras) e regras (a gramática).

mais genérico:

```
00941990-v {proferir, falar, dizer, verbalizar, conversar}
```

Criada originalmente para a língua inglesa (a WordNet de Princeton)[13], existem atualmente iniciativas de diferentes línguas para diferentes wordnets, e um dos desafios é manter o alinhamento entre os *synsets* dessas diferentes línguas. A língua portuguesa tem algumas wordnets ou recursos similares, como OpenWordNet-PT[14], OntoPT[15], PULO[16], TeP[17], PAPEL[18]. (Santos et al. 2010) e (Oliveira et al. 2015) trazem comparações entre os recursos e seus dados podem ser livremente utilizados (veja-se a página de cada projeto para mais informações).

1.2. Pré-processamento: De caracteres a tokens e "palavras"

O texto pode ser visto, inicialmente, como uma sequência de caracteres, ou seja, de símbolos reconhecidos por humanos e computadores. A combinação desses caracteres é que dá forma ao conhecimento por eles representado. Caracteres são combinados em *tokens* (sequências de caracteres delimitados por espaços) que podem ser unidades linguísticas como as palavras.

Na junção de caracteres para a formação de *tokens* uma ferramenta computacional simples e útil são as expressões regulares. Por meio das expressões é possível definir padrões de caracteres capazes de interpretar a sequência de entrada da maneira desejada e extrair dela *trechos* relevantes para o processamento que se deseja. Por exemplo, em um *chatbot*, as expressões regulares podem ser usadas para identificar palavras-chave, como em (1), ou padrões correspondentes a um número de pedido (2) ou a uma data (3)[19]:

(1) `reclamação|quebrado|trocar`

(2) `[A-Z]{3}-[0-9]{3}-[0-9]{3}`

(3) `[0-9]{2}/[0-9]{2}{/[0-9]{4}}?`

Uma das etapas do PLN que se beneficia das expressões regulares é a tokenização. A tokenização (*tokenization*) é o processo de transformar a sequência de caracteres de entrada em unidades que façam sentido para o processamento: os *tokens*. No caso da língua portuguesa, em que o critério gráfico de delimitação costuma funcionar bem[20], palavras são geralmente delimitadas por espaço em branco ou símbolos de pontuação (, - ; : etc.) esse processo não é tão complexo quanto línguas aglutinativas, como o alemão,

[13] <http://wordnet.princeton.edu/>

[14] <http://wn.mybluemix.net/>

[15] <http://ontopt.dei.uc.pt/>

[16] <http://wordnet.pt/>

[17] <http://www.nilc.icmc.usp.br/tep2/index.htm>

[18] <https://www.linguateca.pt/PAPEL/>

[19] Em uma expressão regular o símbolo "|" separa as opções; "[" e "]" delimitam grupos de caracteres; "{" e "}" são usados para agrupar um padrão ou, quando há um número *n* entre eles, *n* indica o número de vezes que o padrão que os antecede se repete e "?" indica que o padrão que o antecede pode ou não ocorrer.

[20] Dissemos "costuma funcionar bem" porque o critério gráfico baseia-se na ortografia, e a ortografia é conhecidamente a dimensão mais arbitrária de uma língua, como vemos em *embaixo* (uma palavra) e *em cima* (duas palavras).

ou nas quais a delimitação não é tão explícita.

No exemplo apresentado na seção 1.1, os *tokens* seriam como os listados a seguir:

Com	estas	palavras	,	André	Coruja	,
além	de	quebrar	o	gelo	que	havia
esfriado	o	clima	,	devolveu	ao	recinto
a	eloquência	necessária	para	que	a	sessão
continuasse	.					

Um conceito relacionado ao de *token* é o *type*. Enquanto *token* se refere a cada ocorrência, o *type* contabiliza a quantidade de *tokens* únicos independente do número de ocorrências no *corpus*. Assim, por exemplo, no exemplo anterior, apesar de termos 30 *tokens*, há apenas 25 *types*, pois os *tokens* "," (3 ocorrências), "o" (2 ocorrências), "que" (2 ocorrências) e "a" (2 ocorrências) são contabilizados apenas uma vez na contagem de *types*. Desse modo, a quantidade de *types* é usada como um indicativo do tamanho do vocabulário de um *corpus* e, consequentemente, de sua cobertura linguística.

Alcançar 100% de cobertura linguística em uma aplicação de PLN é algo utópico, porque o léxico de uma língua é muito amplo e novas palavras surgem em uma velocidade muito maior do que a de atualização dos recursos e modelos. Contudo, muitas das raras e novas palavras compartilham semelhanças com as mais frequentes e já existentes. Essas semelhanças são informações valiosas para se determinar o significado da rara/nova palavra. No exemplo apresentado na seção 1.1 para a palavra *desestender*, a regularidade do uso do elemento *des* como um indicativo de reversão ou negação traz informações valiosas sobre a palavra como um todo.

Considerando essa característica intrínseca das línguas naturais, uma estratégia bastante utilizada pelos modelos de linguagem contextualizados da atualidade é a de dividir uma palavra desconhecida em partes conhecidas no que se acostumou chamar de tratamento de subpalavras (*subwords*). Por exemplo, a técnica de *Byte-Pair Encoding* (BPE), inspirada nas técnicas de compressão de dados, junta os pares de caracteres mais frequentes até ter um vocabulário de tamanho desejado. O tokenizador é treinado com base no conjunto de sequências de caracteres mais frequentes e caracteres restantes, e usado para tokenizar novos textos. Outra técnica de tratamento de subpalavras é o WordPiece usado pelo BERT. Nesse caso, ao invés da frequência, escolhe-se juntar o par de símbolos que maximiza a probabilidade. Seguindo uma abordagem *bottom-up* diferente da dos demais, a Unigram inicializa um vocabulário base com uma grande quantidade de caracteres, sub-palavras e palavras e, durante o treinamento, o vocabulário é progressivamente reduzido, mantendo somente os símbolos mais relevantes, até obter o tamanho desejado.

Subsequente (ou juntamente) à tokenização podem ocorrer outros processamentos que consideram os *tokens* como *lexemas*: a lematização e a radicalização. Um lexema é uma unidade abstrata de significado que corresponde a um conjunto de formas relacionadas. Por exemplo, *bonito*, *bonita*, e *bonitos* são exemplos de lexemas. O *lema* é a forma canônica, dicionarizada, escolhida por convenção para representar um lexema. No caso dos lexemas listados anteriormente, o lema é *bonito*. O *radical* é a palavra básica, sem afixos flexionais, como é o caso de *bonit* para os exemplos de lexemas apresentados

anteriormente.

Esses processamentos podem ser relevantes ou não, depende da aplicação. Para algumas aplicações de PLN não é interessante saber a forma flexionada da palavra (o que aumenta considerável e às vezes desnecessariamente o tamanho do vocabulário), mas sim apenas sua forma canônica (lema) ou raiz. Já para outras aplicações, os resultados são melhores quando se considera todas as formas superficiais (flexionadas) do *corpus* sem quaisquer processamentos adicionais além da tokenização.

No exemplo apresentado na seção 1.1, os lemas correspondentes aos *tokens* retornados pelo processamento com o Spacy[21], uma das ferramentas mais utilizadas para pré-processamento de textos, são[22]:

Com	**este**	**palavra**	,	André	Coruja	,
além	de	quebrar	o	gelo	que	**haver**
esfriar	o	clima	,	**devolver**	ao	recinto
o	eloquência	**necessário**	**parir**	que	**o**	sessão
continuar	.					

Por fim, outro conceito relacionado ao processamento de *tokens* é o de *stopwords*. Em diversas aplicações de PLN é interessante desconsiderar algumas palavras que pouco acrescentam ao conteúdo do texto, como preposições, determinantes, conjunções etc. Essas palavras são conhecidas como *stopwords*. É importante notar que, além das palavras clássicas mencionadas, as *stopwords* podem variar de acordo com o contexto específico com o qual estamos trabalhando. Por exemplo, se estivermos tratando um *corpus* de SMS com o intuito de extrair informações acerca das operadoras e assuntos mais comumente abordados por esse meio de comunicação, termos como "SMS" ou "mensagem" se tornam irrelevantes para a análise, dada sua alta frequência e baixa relevância. Assim, tratá-los como *stopwords* nessa análise específica pode servir como uma boa estratégia para a tarefa/aplicação de PLN em questão.

No exemplo apresentado na seção 1.1, os *tokens* resultantes após desconsiderar as *stopwords* comumente definidas no português, novamente usando o Spacy, seriam como os listados a seguir:

–	–	palavras	–	André	Coruja	–
–	–	quebrar	–	gelo	–	havia
esfriado	–	clima	–	devolveu	–	recinto
–	eloquência	necessária	–	–	–	sessão
continuasse	–					

O código usado para esses processamentos usando o Spacy, e processamentos similares usando o NLTK[23], está disponível em: <https://github.com/brasileiras-pln>

Assim como a utilização do lema ou da forma flexionada depende da aplicação, a eliminação de *stopwords* depende do que se pretende com o texto. *Dar tempo* é diferente de *dar um tempo* e *camisa de força* é diferente de *camisa* e *força* usados isoladamente.

[21] <https://spacy.io/>

[22] Note que o Spacy lematizou equivocadamente a preposição *para* como se fosse o verbo *parir*.

[23] <https://www.nltk.org/>

Além disso, é de suma importância a consideração da língua específica com a qual se está trabalhando e suas idiossincrasias. A maioria dos recursos disponíveis para tratamento e processamento dos *corpora* se baseia principalmente em línguas de maior prestígio socioeconômico, como o inglês, de modo que devemos entender a aplicação de tais recursos e adaptá-los, se necessário, às características da língua com a qual estivermos lidando.

Ressaltamos que os processamentos que apresentamos aqui podem servir como ferramentas bastante úteis para tarefas de PLN. No entanto, devemos sempre lembrar que estamos trabalhando com as línguas humanas, ou seja, com um material que produz significados e sentidos na comunicação e que deve ser entendido com todas as sua complexidades e peculiaridades. Portanto, tais processamentos não devem ser aplicados como uma rotina de tratamento prescritiva e rígida. O trabalho do especialista em PLN com a análise qualitativa dos dados e objetivos é indispensável para a tomada de decisão em cada uma das etapas da construção dos modelos, desde a escolha do *corpus* e seu pré-processamento.

1.3. Representação de palavras: *one-hot-encoding*, *word embeddings* e modelos de linguagem contextualizados

Avançando um pouco mais no nosso entendimento do que é o PLN, nesta seção falaremos sobre as principais formas de representação do conteúdo textual, todas relacionadas ao processamento das unidades linguísticas considerando seus contextos e suas frequências de ocorrência.

O tipo mais tradicional de representação de palavras é o *one-hot encoding*. Nesse tipo de representação, uma palavra é representada em um vetor cuja dimensão é a mesma do tamanho do vocabulário que ela representa. Cada palavra é, então, representada como um vetor binário (contendo 0s e 1s) que tem 1 em apenas uma posição. A Figura 1.1 ilustra essa ideia. Para representar uma sentença, as posições das palavras que ocorrem na sentença recebem o valor 1 enquanto as demais ficam com valor 0.

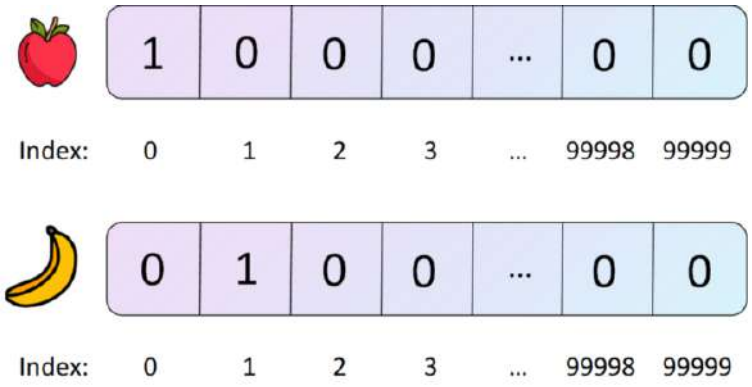

Figura 1.1. Representação vetorial *one-hot encoding*

Fonte: <https://towardsdatascience.com/deep-learning-for-nlp-word-embeddings-4f5c90bcdab5>

Uma das grandes limitações desse tipo de representação é sua esparsidade, uma

vez que, para cada palavra, apenas uma posição será preenchida com 1 desperdiçando, assim, as demais posições que poderiam ser usadas para representar outros traços dessa palavra. Outra limitação é que, em uma representação *one-hot encoding*, a similaridade entre palavras semanticamente próximas como *maçã* e *banana* não é facilmente capturada pois o contexto de ocorrência similar entre elas não é representado no vetor.

A necessidade de considerar o contexto na geração das representações de palavras surgiu com a *hipótese distributiva* formulada na década de 1950 por Joss (1950), Harris (1954) e Firth (1957) apud (Jurafsky e Martin 2021). De acordo com essa hipótese, assume-se que palavras semelhantes ocorrem em contextos similares. Foi com base nesta hipótese que surgiram representações vetoriais (*embeddings* ou vetores de palavras).

Os primeiros modelos vetoriais (também conhecidos como modelos distribucionais) se baseavam em matrizes de co-ocorrência que relacionam ocorrências de termos com documentos ou outros termos, como: (i) a matriz de frequência termo-documento, que associa a frequência de ocorrência de termos em documentos (sentenças); (ii) a matriz de frequência termo-termo, que associa a frequência de ocorrência entre termos e (iii) a matriz TF-IDF, que representa um termo em função da relação de sua frequência de ocorrência (TF, *term frequency*) e a frequência inversa de documentos nos quais ele ocorre (IDF, *inverse document frequency*). Embora tenham trazido avanços na representação ao se considerar o contexto de ocorrência das palavras, esses modelos vetoriais tradicionais ainda são esparsos (contêm muitos 0s) e não escaláveis (à medida que o número de documentos e o vocabulário cresce, a dimensão das matrizes torna-se um gargalo).

Como alternativa, surgiram as formas de representação contínuas (também conhecidas como vetores densos) que são vetores numéricos de dimensão menor do que o tamanho vocabulário (geralmente entre 50 e 1000) com valores reais em cada uma dessas posições. Por meio desses vetores é possível realizar cálculos espaciais facilitados para encontrar a similaridade entre palavras. Essas representações têm sido as mais utilizadas na atualidade devido aos bons resultados em aplicações de PLN que, segundo (Jurafsky e Martin 2021), se devem a sua capacidade de generalização (evitando o *overfitting*) e captura de sinônimos. Tais representações podem ser divididas em: estáticas (mais conhecidas como *word embeddings*) e dinâmicas (modelos de linguagem contextualizados). Enquanto os modelos de representação estática aprendem uma *embedding* fixa para cada palavra no vocabulário, os modelos dinâmicos geram um vetor para cada contexto de ocorrência das palavras.

Assim como as demais representações vetoriais, as *word embeddings* são aprendidas a partir de um grande *corpus* por meio de métodos de contagem, como o GloVe (Pennington et al. 2014), ou neurais, como o Word2Vec (Mikolov et al. 2013) e o FastText (Bojanowski et al. 2016). Na geração das *word embeddings*, os contextos de ocorrência das palavras vão sendo considerados para se calcular os valores numéricos que populam cada uma das *n* dimensões usadas na representação. Idealmente, é como se cada uma das dimensões representasse algum traço semântico/sintático da palavra sendo representada. A Figura 1.2 ilustra essa ideia.

Neste exemplo, traços semânticos – ser vivo (*living being*), felino (*feline*), humano (*human*) e realeza (*royalty*) – e sintáticos – gênero (*gender*), verbo (*verb*) e plural – aparecem como significado idealizado para as dimensões usadas na representação das

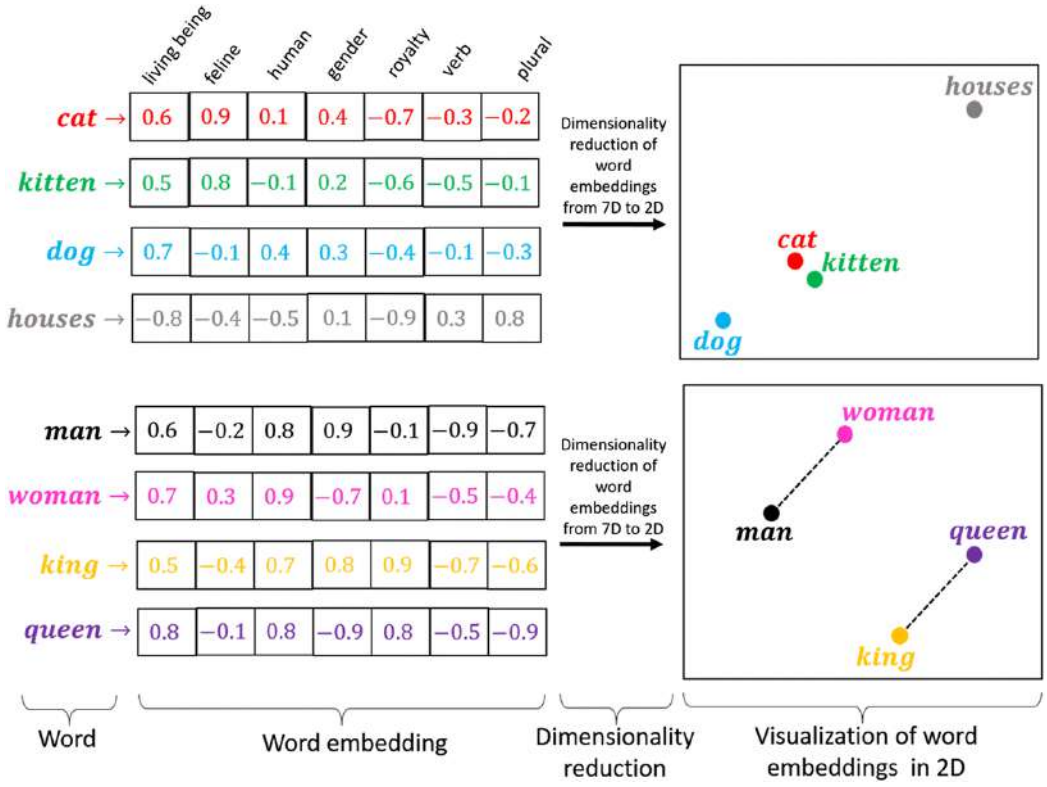

Figura 1.2. Representação vetorial *word embedding*

Fonte: <https://medium.com/@hari4om/word-embedding-d816f643140>

palavras *gato* (*cat*), *gatinho* (*kitten*), *cachorro* (*dog*) e *casas* (*houses*). Com base na distância espacial entre as representações dessas palavras é possível recuperar a similaridade entre palavras próximas, como *gato* e *gatinho*, por exemplo. Vale ressaltar, contudo, que na prática não é possível saber ao certo o que está sendo representado ou capturado por cada uma das dimensões das *word embeddings*.

Códigos para manipulação e criação de *word embeddings* estão disponíveis em: <https://github.com/brasileiras-pln>

Embora tenham representado um grande avanço na área de PLN, as *word embeddings* também têm limitações. A principal delas é conhecida como confluência de significados. Uma vez que apenas uma representação é gerada para a mesma forma superficial de uma palavra, os diversos significados dessa mesma forma superficial são "misturados" na mesma *embedding* perdendo-se, assim, informação linguística valiosa.

Assim, em 2017, com o surgimento da arquitetura neural *transformer* proposta inicialmente para modelos de tradução automática, diversos modelos de linguagem contextualizados (ou vetores de palavras dinâmicos ou *embeddings* contextualizadas) foram propostos. Nesses modelos, cada palavra é representada por um vetor diferente cada vez que um contexto de ocorrência diferente é detectado, ou, como bem colocado por (Jurafsky e Martin 2021): "enquanto os vetores estáticos representam os significados dos *types* (entradas do vocabulário), os vetores dinâmicos representam os significados dos *tokens* (instâncias de um *type* um contexto específico)". São tantas as arquiteturas neurais

propostas para gerar esses modelos que a Figura 1.3 provavelmente já está defasada.

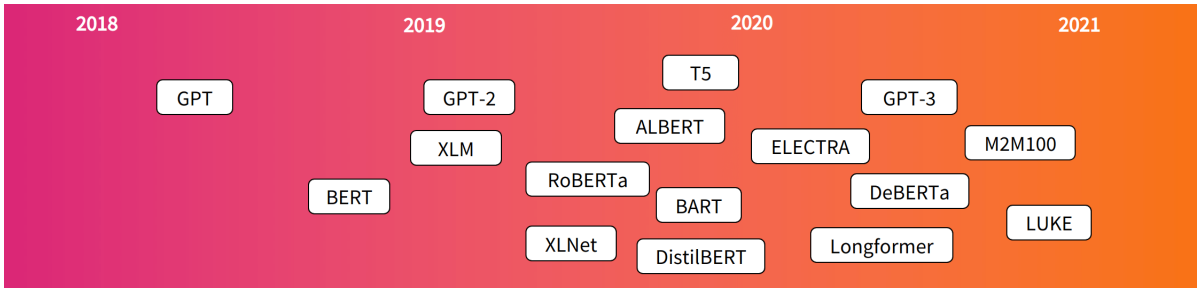

Figura 1.3. Linha do tempo dos modelos *transformers* mais populares em Março 2022

Fonte: <https://huggingface.co/blog/bert-101>

O PLN usando modelos de linguagem contextualizados envolve duas etapas: pré-treinamento e refinamento ou ajuste fino (*fine-tuning*). No pré-treinamento, imensas quantidades de dados e muito poder computacional (em máquinas dotadas de várias GPUs) são usados para gerar um modelo computacional capaz de abstrair características da linguagem. O pré-treinamento é, portanto, o processo de aprender algum tipo de representação de significado das palavras (ou sentenças) a partir do processamento de grandes *corpora* (Jurafsky e Martin 2021). Esse pré-treinamento é realizado de modo auto-supervisionado usando estratégias como *Masked Language Modeling* (MLM) ou *Next Sentence Prediction* (NSP).

No MLM o objetivo é prever uma palavra que foi artificialmente substituída por um *token* especial: `[MASK]`. Por exemplo, considerando o exemplo da seção 1.1, a palavra *quebrar* poderia ser mascarada como apresentado a seguir:

> Com estas palavras, André Coruja, além de `[MASK]` o gelo que havia esfriado o clima ...

Assim, a partir de um grande *corpus*, alguns *tokens* são selecionados aleatoriamente para serem mascarados, outros são substituídos por *tokens* aleatórios e os demais mantidos sem alteração. No BERT, 15% dos *tokens* no *corpus* de treinamento são selecionados para esse processo. Destes, 80% são mascarados, 10% são substituídos por *tokens* aleatórios e 10% permanecem inalterados (Jurafsky e Martin 2021). O objetivo da estratégia de MLM é tenta prever qual é a palavra que foi mascarada considerando os contextos de ocorrência tanto à esquerda quanto à direita (*bidirectional encoder*). Essa mesma ideia pode ser estendida para mascarar sequências de *tokens* como ocorre no SpanBERT (Joshi et al. 2020).

Enquanto o foco da estratégia de treinamento MLM é predizer as palavras com base em suas vizinhanças (contexto à esquerda e à direita), a NSP visa prever a próxima sentença. Essa estratégia é bastante útil para aplicações de PLN nas quais o relacionamento entre as sentenças é importante, como na detecção de paráfrases. Na NSP, o pré-treinamento é feito a partir de pares de sentenças que estão relacionadas entre si (como sentenças adjacentes) ou não. Segundo (Jurafsky e Martin 2021), no BERT, 50% dos pares de treinamento são positivos e os outros 50% são pares gerados aleatoriamente.

Nesse caso, o objetivo é dizer se o par de sentenças é um par positivo ou não. Para tanto, dois novos *tokens* especiais são usados: `[CLS]`, que indica o início do par de sentenças; e `[SEP]`, que é usado para separar as sentenças e marcar o final do par. Adaptando o exemplo da seção 1.1, poderíamos ter algo do tipo:

> `[CLS]` Com estas palavras, André Coruja, além de quebrar o gelo que havia esfriado o clima, `[SEP]` devolveu ao recinto a eloquência necessária para que a sessão continuasse. `[SEP]`

Vale mencionar que essas estratégias são de aprendizado não supervisionado, ou seja, os dados de entrada são dados puros (brutos), sem qualquer tipo de anotação (rotulação) da tarefa de PLN que se deseja aprender/modelar. Diversos modelos de linguagem pré-treinados estão disponíveis no Hugging Face[24].

A partir de um modelo pré-treinado o que se faz é realizar o ajuste fino (*fine-tuning*) para uma tarefa específica usando um conjunto menor de dados rotulados. Para (Jurafsky e Martin 2021), o *fine-tuning* é um processo que continua (estende) o pré-treinamento de um modelo de linguagem contextualizado, geralmente adicionando camadas neurais de classificação e/ou ajustando parâmetros, a fim de realizar alguma tarefa fim como etiquetagem morfossintática ou análise de sentimentos. Exemplos de *fine-tuning* de modelos pré-treinados para o português do Brasil, em tarefas de PLN, são apresentados na seção 1.5.

1.4. Aprendizado supervisionado

Nesta seção falaremos de uma importante parte do PLN que é o enriquecimento do conjunto de treinamento (*corpus*) por meio da anotação. A anotação é feita para evidenciar algum fenômeno linguístico que se deseja identificar/processar automaticamente por meio de um modelo computacional.

1.4.1. Anotação de *corpus* e a criação de *datasets* linguísticos

Chamamos de *datasets* (linguísticos), ou *corpora* anotados padrão ouro, a uma coleção de documentos (textos) anotados. Anotar um texto, por sua vez, significa adicionar informação linguística a palavras ou porções de textos maiores. Esta informação linguística pode ser de natureza bastante variada, e alguns tipos mais comuns de anotação são:

- classes de palavras (chamada anotação de POS, do inglês *part of speech*): anotação que atribui a cada palavra uma etiqueta do tipo substantivo, verbo, adjetivo etc.
- sintaxe: anotação que atribui a cada palavra uma etiqueta do tipo sujeito, objeto direto, adjunto adverbial etc.
- polaridade: anotação que atribui a cada palavra ou conjunto de palavras uma etiqueta do tipo positivo, negativo ou neutro conforme o seu valor semântico/cultural: *defeito*, *cabeça dura*, *quebrar* e *doença* são negativos; *amor*, *perfeito*, *adorar* e *inteligente* são positivos, e *surpresa*, *computador* e *normal* são neutros.

[24]<https://huggingface.co/blog/bert-101>

- entidades nomeadas: anotação que atribui a cada palavra ou conjunto de palavras uma etiqueta semântica, que pode ser genérica (como PESSOA, LUGAR, ORGANIZAÇÃO) ou específica (como ROCHA para termos como anidrita, basalto e folhelho considerando entidades do domínio geológico).

- correferência: anotação que relaciona duas formas que se referem à mesma entidade. Na frase abaixo[25] os índices *i* e *ii* sinalizam as correferências, que podem estar restritas a uma frase ou levar em conta parágrafos ou o texto inteiro

  > A tenista americana *Serena Williams*$_i$, de 40 anos, anunciou na terça-feira (9) que vai se aposentar das quadras [...]. Apesar de não deixar claro a data de aposentadoria, *ela*$_i$ sugeriu que o *US Open*$_{ii}$, que ocorre entre 29 de agosto e 11 de setembro deste ano, pode ser o último *torneio*$_{ii}$ da sua vitoriosa carreira profissional.

- anotação de relações: similar à anotação de correferência, mas leva em conta uma variedade de relações semânticas (inclusão, localização etc), e não apenas a relação de *identidade*;

- anotação de similaridade semântica: anotação que indica o grau de semelhança entre duas porções de um texto – por exemplo, entre duas frases. Em geral, o grau de semelhança pode ser de três tipos:

  1. acarretamento - o sentido de uma frase está incluído no sentido de outra.
  2. contradição - o sentido de uma frase contradiz o sentido de outra.
  3. neutro - a frase hipótese pode ser verdadeira dada a frase premissa.

Datasets codificam o desempenho humano em uma determinada tarefa. Por isso, para que um *dataset* seja útil no aprendizado automático, é importante que a anotação nele contida tenha sido feita (ou revista) por pessoas. Por isso, também é possível fazer referência a este tipo de material como um *corpus* padrão-ouro (*gold standard*).

1.4.2. Relevância de datasets no aprendizado supervisionado

Um dataset – ou *corpus* anotado – é um recurso linguístico. E sua relevância no aprendizado supervisionado se deve a três motivos.

O aprendizado de máquina precisa de exemplos do que será aprendido. E apesar dos avanços das redes neurais e do aprendizado profundo, bons exemplos continuam necessários, com a vantagem de agora ser possível oferecer aos algoritmos uma quantidade menor de exemplos, graças ao ajuste fino (*fine-tuning*) apresentado na seção 1.3.

O segundo motivo que torna *datasets* fundamentais em diversas tarefas de PLN é que eles facilitam o processo de avaliação de um sistema e de comparação entre sistemas. Isto porque se a anotação codifica, no *corpus*, a compreensão humana sobre algo, e o que queremos das máquinas em certas tarefas é que elas reproduzam esta compreensão

[25] O trecho foi retirado de <https://www.nexojornal.com.br/expresso/2022/08/11/O-que-Serena-Williams-fez-que-nenhum-outro-tenista-conseguiu>

humana, então a melhor maneira de saber o quão bom é um resultado é comparando-o com o entendimento humano.

Por fim, tecnologias /algoritmos podem ser independentes de língua, mas recursos linguísticos, não. Ou seja, para garantir um bom aprendizado em uma tarefa de identificação de entidades, por exemplo, adianta pouco treinar o modelo que seja efetivo para a língua portuguesa em um *dataset* criado para uma língua diferente do português.

1.4.3. Criação de um *dataset* linguístico: planejamento e esquema de anotação

Um dos primeiros passos na criação de um *dataset* linguístico é a definição do conjunto de etiquetas que será usado na anotação. O conjunto de etiquetas de anotação é chamado *tagset*, e subjacente a um *tagset* há um esquema de anotação, isto é, uma forma de descrever e organizar as classes (etiquetas) que o compõem.

O conjunto de etiquetas pode vir diretamente de uma teoria ou pode apenas se inspirar em algum modelo teórico; pode ser a simplificação de uma teoria ou a convergência de diferentes teorias que se debruçam sobre o mesmo fenômeno. Além disso, as etiquetas podem ser criadas tendo em vista uma determinada tarefa/aplicação em mente, como a anotação de polaridades e a anotação de entidades. Assim, a anotação pode codificar tanto aspectos estritamente linguísticos (como classes de palavras) como aspectos mais genéricos e dependentes de uma boa capacidade de interpretação de textos (caso da anotação de polaridades) ou o conhecimento específico de algum domínio, como geologia, direito ou medicina. Neste caso, a participação de especialistas das respectivas áreas nos projetos de anotação é crucial.

Excetuando-se o cenário em que todas as categorias de anotação já estão dadas de antemão por alguma teoria, e as situações em que se deseja replicar (para uma outra língua ou para um outro *corpus*) uma determinada anotação que já existe, nos demais contextos será necessário criar um esquema de anotação.

Definir um conjunto de etiquetas (um conjunto de classes) e sua forma de aplicação reflete uma maneira de ver a tarefa. Por isso, quanto mais bem definido o problema (a tarefa), mais chances de sucesso. Caso seja necessário criar um esquema de anotação, devemos logo responder às seguintes perguntas:

- Qual o objetivo da anotação?
- A que ela serve?

Um aspecto importante de um esquema de anotação é que ele favoreça a generalização. A terminologia de uma área, embora contenha termos específicos daquele domínio, não é exatamente um conjunto de entidades desse mesmo domínio: é importante que esses diferentes termos sejam agrupados em classes mais amplas, ou seja, os termos podem ser instâncias de categorias de anotação. Por exemplo, "conglomerado", "lamito" e "arenito" são termos da Geologia. Na anotação da frase (a) precisaremos de uma classe ampla que os agrupe e generalize – por exemplo, uma classe ROCHA.

> A formação é constituída por conglomerados, arenitos conglomeráticos, arenitos e lamitos.

O processo de anotação segue as seguintes etapas:

1. Levantamento bibliográfico sobre o que já existe relacionado à questão, em termos teóricos (descrição linguística, por exemplo) e aplicados (existem anotações do mesmo tipo, ou diretamente relacionadas? Quais os problemas enfrentados?);
2. Elaboração de um esquema ou modelo de anotação, que contém as primeiras generalizações acerca do fenômeno observado, isto é, a primeira proposta de etiquetas (categorias);
3. Aplicação dessas etiquetas a uma amostra mais ampla;
4. Refinamento progressivo do esquema de anotação;
5. Observação dos casos em que as generalizações não se aplicam (com a ressalva de que as irregularidades devem estar igualmente marcadas no *corpus*).

Com relação às etiquetas, as seguintes dicas também podem ajudar:

1. A existência de uma etiqueta do tipo MISCELÂNEA, ou OUTROS, é útil para os casos não previstos. Esta etiqueta pode incluir casos que serão especificados no futuro;
2. Admitir a indeterminação, isto é, que duas ou mais etiquetas estejam igualmente adequadas, no mesmo contexto. A possibilidade de anotação múltipla permitiria que as etiquetas AVISO, AMEAÇA e CONSELHO sejam atribuídas à frase *Se eu fosse você, deixaria a cidade imediatamente*, por exemplo.

1.4.4. Avaliação da anotação

No PLN, temos duas maneiras de aferir a qualidade das anotações: comparar entre si as anotações produzidas pelos anotadores (humanos), ou comparar essas anotações e um gabarito (e este gabarito foi produzido por alguém). Como nem sempre existe um gabarito à disposição, a comparação entre análises de diferentes anotadores acaba sendo a solução adotada. Assim, a ideia de uma anotação correta (supostamente fornecida pelo gabarito) é substituída pela ideia de uma anotação consistente (todos os anotadores analisaram os fenômenos da mesma maneira). Este procedimento de avaliação e verificação da anotação humana é chamado de Concordância Inter-Anotadores (*Inter annotator agreement*) (Artstein 2017).

1.4.5. Considerações finais sobre o processo de anotação

Em resumo, um projeto cuidadoso de anotação deverá levar em conta:

- Clareza quanto ao fenômeno que será anotado (que se reflete em um bom esquema de anotação);
- Escolha do *corpus* adequado (um *corpus* composto por relatórios de pesquisa é pouco adequado para a anotação de ironia, por exemplo);

- um conjunto de etiquetas e um esquema de anotação;
- instruções para a aplicação das etiquetas (documentação), que contenham tanto os casos gerais quanto as exceções;
- Avaliação da anotação (concordância entre anotadores);
- Verificação quanto à eficiência da anotação (por exemplo, o tempo levado e o nível de treinamento/conhecimento necessário por parte de quem vai anotar).

1.5. Aplicações de PLN com modelos pré-treinados

Por fim, esta última seção apresenta o *fine-tuning* de modelos computacionais estado-da-arte para o português em algumas aplicações de PLN.

1.5.1. Análise de sentimentos em avaliações de produtos da B2W (Americanas s.a.)

Uma das aplicações mais conhecidas do PLN é a análise de sentimentos (ou mineração de opiniões) que visa extrair o sentimento (ou a polaridade) em sentenças que expressam opiniões. A classificação, neste caso, pode ser em emoções (gratidão, alívio, remorso, etc.) como ocorre no GoEmotions (Demszky et al. 2020) ou em polaridade/valência (positiva, negativa ou neutra).

Nesta seção, vamos ilustrar o uso de um modelo pré-treinado para o português, o BERTimbau[26] (Souza et al. 2020), na análise de polaridade/valência usando o *corpus* B2WReviews[27], um recurso disponível livremente que contém mais de 130.000 avaliações (*reviews*) de clientes sobre produtos vendidos pela B2W (atual Americanas S.A.). De acordo com as informações na página deste *corpus*, essas avaliações foram coletadas do site Americanas.com nos primeiros 5 meses de 2018. A Figura 1.4 ilustra um trecho desse *corpus* com os campos de informação que serão usados no ajuste do modelo: `review_text` (texto da avaliação escrita pelo usuário), `overall_rating` (quantidade de 1 a 5 estrelas atribuídas pelo usuário para o produto adquirido) e `polaridade` (novo campo inserido neste exemplo para mapear `overall_rating` para polaridades negativa, neutra ou positiva).[28]

A partir das informações textuais (textos das avaliações) o modelo pré-treinado do BERTimbau foi usado no *fine-tuning* para classificação em três classes possíveis (polaridade): positiva, negativa ou neutra. Para determinar essas classes, fizemos o mapeamento da nota atribuída pelos clientes (`overall_rating`) aos produtos que adquiriram como: 4 e 5 (positiva), 3 (neutra) e 1 e 2 (negativa). Desse modo, vale ressaltar que a anotação dos dados foi realizada pelos próprios autores das avaliações não sendo, portanto, uma anotação *gold standard* realizada seguindo critérios rigorosos.

A análise de sentimentos é um exemplo de tarefa de classificação de uma sequência (*sequence classification*), uma vez que toda a sequência de entrada é processada para

[26]<https://github.com/neuralmind-ai/portuguese-bert>

[27]<https://github.com/b2w-digital/b2w-reviews01>

[28]É importante mencionar que os textos são textos gerados pelos usuários e portanto, podem conter abreviações comuns na linguagem coloquial (como *td* e *hj*), ausência de acentos (como *maos*) e erros de grafia (como em *ofereçe*), entre outros.

review_text	overall_rating	polaridade
A TV chegou em minha casa dia 21/12/2017 liguei todos os fios e td certo, chegou hj dia 01/01/2018 a TV não quer maos ligar, nem acende o led, lamentável acaba a confiança na marca!!! Não recomendo.	1	negativa
a mochila nao esta fechando direito por isso nao recomendo se meu filho nao tivesse deixado suja ia devolver	2	negativa
Produto ofereçe o que prometeu no anúncio, entrega muito rapida bem antes do prazo	3	neutra
Estou contente com a compra entrega rápida o único problema com as Americanas é se houver troca ou devolução do produto o consumidor tem problemas com espera.	4	positiva
Excelente produto, por fora em material acrílico super resistente e por dentro em adamantio, faz milagre com qualquer bebida. Sugiro aproveitarem a promoção antes que acabe.	5	positiva

Figura 1.4. Trecho do *corpus* B2WReviews apresentando apenas as informações utilizadas no treinamento

que seja gerada a saída (neste caso, a polaridade). No BERT, o *token* especial `[CLS]` é colocado no início de todas as sequências gerando representações para elas (y_{CLS}) e os pesos da rede (W_C) são ajustados para que o vetor de saída gerado represente a qual das classes uma dada sequência de entrada está associada. Desse modo, para classificar uma nova sequência de entrada, ela é processada pelo modelo pré-treinado para gerar a representação correspondente (y_{CLS}) que é, então, associada à classe correspondente com base nos pesos aprendidos durante o treinamento (W_C). O ajuste fino dos valores em W_C requer um treinamento supervisionado com base em sequências de entrada rotuladas com a classe correta.

No nosso caso, depois de 3 épocas de treinamento, usando um pouco menos de 8.000 instâncias do *corpus*, o modelo ajustado alcançou os resultados apresentados na Figura 1.5 para o conjunto de teste (com cerca de 2.000 instâncias). De acordo com os valores é possível notar que o modelo se sai muito bem na classificação das classes positiva e negativa, com precisões próximas a 90%, mas tem um desempenho bem pior na classificação da classe neutra, com apenas 44% de precisão. No geral, o valor da média harmônica entre precisão e cobertura (F1) foi de 84%.

Embora não seja o mesmo tipo de texto (avaliações de produtos), por curiosidade verificamos qual a polaridade que esse modelo gera para a sentença de exemplo da seção 1.1. O resultado é apresentado na Figura 1.6.

```
              precision    recall  f1-score   support

    Negativa       0.88      0.90      0.89       476
      Neutra       0.44      0.35      0.39       240
    Positiva       0.90      0.93      0.91      1228

    accuracy                           0.85      1944
   macro avg       0.74      0.73      0.73      1944
weighted avg       0.84      0.85      0.84      1944

F1:  0.8420356252433361 Accuracy:  0.8482510288065843
```

Figura 1.5. Resultados do *fine-tuning* para a análise de polaridade

texto	**polaridade**
Com estas palavras, André Coruja, além de quebrar o gelo que havia esfriado o clima, devolveu ao recinto a eloquência necessária para que a sessão continuasse.	positiva

Figura 1.6. Polaridade atribuída à sentença de exemplo da seção 1.1 usando o modelo ajustado com o B2WReviews

1.5.2. Etiquetagem morfossintática usando o *corpus* Mac-Morpho

Além da categorização textual realizada para sentenças (*sequence classification*), apresentada na seção anterior, outra aplicação bastante comum em PLN é a categorização de *tokens* (*sequence labelling*). Para ilustrar tal aplicação, nesta seção será descrito o uso do BERTimbau para a etiquetagem morfossintática (*part-of-speech tagging*). Para tanto, o *corpus* utilizado é o Mac-Morpho[29] (Aluísio et al. 2003). A Figura 1.7 ilustra uma sentença desse *corpus* acompanhada de suas etiquetas morfossintáticas.[30]

O Mac-Morpho é um *corpus* de textos em português do Brasil anotados com etiquetas morfossintáticas (*part-of-speech tags*). Ele é composto por 1,1 milhões de palavras derivadas de artigos do jornal brasileiro Folha de São Paulo[31]. Como tal, esse *corpus*, diferente do usado na seção anterior, possui textos que seguem a norma culta da língua portuguesa e não apresentam erros ortográficos e gramaticais. A anotação deste *corpus* foi realizada inicialmente de modo automático pelo *parser* PALAVRAS (Bick 2000) e, então, revisada por 4 anotadores.

Diferente do processo de classificação de sequência (adotado no *fine-tuning* para a análise de sentimentos), aqui o vetor de saída está associado a cada *token* e não mais a uma sequência completa. Depois de 3 épocas de treinamento o modelo ajustado alcançou os resultados apresentados na Figura 1.8 para o conjunto de teste, ou seja, uma acurácia quase perfeita de 98%. Aqui vale ressaltar que o desempenho dos modelos computacionais varia muito entre as tarefas de PLN, uma vez que tarefas mais objetivas como a etiquetagem morfossintática de *tokens* tendem a ser mais fáceis de generalizar do que as

[29] <http://nilc.icmc.usp.br/macmorpho/>

[30] A lista completa as etiquetas utilizadas na anotação desse *corpus* está disponível em: <http://nilc.icmc.usp.br/macmorpho/macmorpho-manual.pdf>.

[31] <https://www.folha.uol.com.br/>

Ainda ADV	em PREP	dezembro N	de PREP	1990 N	, PU
foi V	editada PCP	a ART	famosa ADJ	289 N	, PU
que PRO-KS	modificava V	a ART	sistemática N	da PREP+ART	arrecadação N
do PREP+ART	ITR NPROP	e KC	alterava V	suas PROADJ	alíquotas N
. PU					

Figura 1.7. Trecho do *corpus* Mac-Morpho onde cada *token* está associado a uma etiqueta morfossintática

tarefas mais subjetivas como a análise de sentimentos.

A título de ilustração, o modelo ajustado gerou as etiquetas de *part-of-speech* para a sentença de exemplo da seção 1.1 como apresentado na Figura 1.9.

```
              accuracy                           0.98     33729
             macro avg       0.96      0.94      0.95     33729
          weighted avg       0.98      0.98      0.98     33729
```

Figura 1.8. Resultados do *fine-tuning* para a etiquetagem morfossintática

Com PREP	estas PROADJ	palavras N	, PU	André NPROP	Coruja NPROP	, PU
além PREP	de PREP	quebrar V	o ART	gelo N	que PRO-KS	havia V
esfriado PCP	o ART	clima N	, PU	devolveu V	ao PREP+ART	recinto N
a ART	eloquência N	necessária ADJ	para PREP	que KS	a ART	sessão N
continuasse V	. PU					

Figura 1.9. Etiquetas geradas para a sentença de exemplo da seção 1.1 usando o modelo ajustado com o Mac-Morpho

1.5.3. Tradução automática en-pt usando TED *corpus*

A última aplicação de *fine-tuning* de um modelo pré-treinado envolvendo o português do Brasil apresentada neste documento é a de tradução automática. Neste caso, a partir de pares de sentenças paralelas (sentenças que são a tradução umas das outras), o modelo pré-treinado é ajustado para ser capaz de gerar uma sequência de *tokens* alvo (de saída) a partir de uma sequência de *tokens* fonte (de entrada).

Para tanto, aqui optamos por partir de um modelo pré-treinado do T5[32] e realizar um ajuste fino com o TED[33] *corpus* para o qual pares de sentenças paralelas português-inglês são apresentados na Figura 1.10.

Sentença em português	**Sentença em inglês**
Este é o caminho para tratar as pessoas como adultos responsáveis.	This is the way we treat people as responsible adults.
Ela morava na rua havia mais de 20 anos, tinha problemas mentais e era uma alcoólatra inveterada.	She had been on the street for 20-plus years, had mental health issues and was a severe alcoholic.
Isso é na antiga União Soviética entre o Cazaquistão e o Usbequistão, um dos maiores mares interiores do mundo.	This is in the former Soviet Union in between Kazakhstan and Uzbekistan, one of the great inland seas of the world.
Há muitos anos, eu comecei a usar o termo "interseccionalidade" para lidar com o fato de que muitos de nossos problemas de justiça social, como racismo e sexismo, frequentemente se sobrepõem, criando múltiplos níveis de injustiça social.	Many years ago, I began to use the term "intersectionality" to deal with the fact that many of our social justice problems like racism and sexism are often overlapping, creating multiple levels of social injustice.
Havia isso onde todas essas coisas em minha mente podiam estar nesse lugar lá fora, grande e elétrico, existir e escapar, e eu sabia, a partir daí, que eu queria fazer isso pelo resto da vida, fosse algo remunerado ou não. Eu tinha essa paixão e precisava de ferramentas.	There was this place where all these things in my head could go into the great, electric somewhere-out-there and exist and escape, and I knew from this moment on, I wanted to do this for the rest of my life, whether I was going to get paid for it or not.

Figura 1.10. Pares de sentenças paralelas do TED *corpus*

Neste caso, usamos um modelo de linguagem de sequência-para-sequência (ou `seq2seq`) que é o indicado para tarefas de PLN nas quais a entrada é uma sequência de *tokens* e a saída também é uma sequência de *tokens*, mas de tamanho arbitrário; e não mais uma classificação como nos casos anteriores. Assim, modelos pré-treinados `seq2seq` são úteis para aplicações como a tradução automática, a sumarização automática e a geração de legendas para imagens.

Depois de três épocas de treinamento, o modelo ajustado apresentou um BLEU[34] (Papineni et al. 2002) de 0,46 que trouxe uma tradução bastante razoável para a sentença de exemplo da seção 1.1, como ilustrado na Figura 1.11. Neste caso, fazendo uma comparação da saída de nosso modelo ajustado com aquela gerada pelo Google tradutor, temos apenas duas palavras diferentes (destacadas em negrito): o sobrenome Oruja, que nosso modelo traduziu de modo equivocado (o correto seria manter o original); e o termo *re-*

[32]<https://huggingface.co/Helsinki-NLP/opus-mt-ROMANCE-en>

[33]<https://object.pouta.csc.fi/OPUS-TED2020/v1/tmx/en-pt_br.tmx.gz>

[34]O BLEU é uma medida de avaliação automática baseada na co-ocorrência de n-gramas (sequências de n *tokens*) em comum entre a saída gerada por um sistema automático e uma ou mais sentenças de referência.

cinto traduzido para *compound* pelo nosso modelo e para *venue* (uma tradução melhor) pelo Google.

Fonte	Com estas palavras, André Coruja, além de quebrar o gelo que havia esfriado o clima, devolveu ao recinto a eloquência necessária para que a sessão continuasse.
Alvo (nosso modelo)	With these words, André **Oruja**, in addition to breaking the ice that had cooled the climate, returned to the **compound** the eloquence necessary for the session to continue.
Alvo (Google)	With these words, André **Coruja**, in addition to breaking the ice that had cooled the climate, returned to the **venue** the eloquence necessary for the session to continue.

Figura 1.11. Tradução gerada para a sentença de exemplo da seção 1.1 usando o modelo ajustado com o TED *corpus* e o Google tradutor

1.6. Considerações finais

O processamento automático das línguas naturais (ou PLN) é uma área interdisciplinar que tem as línguas humanas como objeto de estudo e processamento. Os processamentos realizados com as técnicas, ferramentas e recursos do PLN podem ser o produto final (como em corretores ortográficos, tradutores e sumarizadores automáticos, ferramentas de auxílio à escrita, etc.) ou o produto intermediário para enriquecimento de dados que poderão ser utilizados em uma aplicação final (como ocorre com os etiquetadores morfossintáticos e os analisadores sintáticos e semânticos, por exemplo).

Saber quanto do PLN é necessário para uma aplicação computacional exige conhecimento das características e peculiaridades da língua natural sendo processada (**conhecimento da língua**), bem como do produto final que se deseja desenvolver (**propósito**). Hoje, com a disponibilização livre de códigos, ferramentas computacionais, recursos linguísticos e modelos pré-treinados, implementar soluções de PLN se tornou uma tarefa bem menos custosa. Dados também estão cada vez mais disponíveis, principalmente aqueles gerados pelas pessoas comuns, em suas comunicações e interações corriqueiras em redes sociais e postagens online. Contudo, códigos e dados sozinhos não são capazes de produzir uma aplicação de PLN robusta. O conhecimento da língua e do propósito para seu processamento devem sempre ser a base de qualquer desenvolvimento em PLN.

Referências

Aluísio et al. 2003 ALUÍSIO, S. et al. An account of the challenge of tagging a reference corpus for brazilian portuguese. In: MAMEDE, N. J. et al. (Ed.). *Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 110–117. ISBN 978-3-540-45011-5.

Artstein 2017 ARTSTEIN, R. Inter-annotator agreement. In: *Handbook of linguistic annotation*. [S.l.]: Springer, 2017. p. 297–313.

Aziz e Specia 2011 AZIZ, W.; SPECIA, L. Fully automatic compilation of a Portuguese-English parallel corpus for statistical machine translation. In: *STIL 2011*. Cuiabá, MT: [s.n.], 2011.

Bick 2000 BICK, E. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Århus: University of Arhus, 2000.

Bojanowski et al. 2016 BOJANOWSKI, P. et al. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

Brown et al. 2020 BROWN, T. et al. Language models are few-shot learners. In: LAROCHELLE, H. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>.

Demszky et al. 2020 DEMSZKY, D. et al. GoEmotions: A dataset of fine-grained emotions. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020. p. 4040–4054. Disponível em: <https://aclanthology.org/2020.acl-main.372>.

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

Joshi et al. 2020 JOSHI, M. et al. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, v. 8, p. 64–77, 01 2020. ISSN 2307-387X. Disponível em: <https://doi.org/10.1162/tacl_a_00300>.

Jurafsky e Martin 2021 JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*. [s.n.], 2021. Draft of December 29, 2021. Disponível em: <https://web.stanford.edu/~jurafsky/>.

Kilgarriff 1997 KILGARRIFF, A. I don't believe in word senses. *Computers and the Humanities*, Springer, v. 31, n. 2, p. 91–113, 1997. ISSN 00104817.

Liu et al. 2019 LIU, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. Cite arxiv:1907.11692. Disponível em: <http://arxiv.org/abs/1907.11692>.

McShane e Nirenburg 2021 MCSHANE, M.; NIRENBURG, S. *Linguistics for the Age of AI*. The MIT Press, 2021. ISBN 9780262363136. Disponível em: <https://doi.org/10.7551/mitpress/13618.001.0001>.

Mikolov et al. 2013 MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, v. 2013, 01 2013.

Oliveira et al. 2015 OLIVEIRA, H. G. et al. As wordnets do português. *Oslo Studies in Language*, v. 7, n. 1, p. 397–424, 2015.

Papineni et al. 2002 PAPINENI, K. et al. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002. p. 311–318. Disponível em: <https://aclanthology.org/P02-1040>.

Pennington et al. 2014 PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Disponível em: <https://aclanthology.org/D14-1162>.

Santos et al. 2010 *Relações semânticas em português: comparando o TeP, o MWN.PT, o Port4NooJ e o PAPEL*. [S.l.]: Associação Portuguesa de Linguística: Lisboa, 2010. 681–700 p.

Souza et al. 2020 SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). *Intelligent Systems*. Cham: Springer International Publishing, 2020. p. 403–417. ISBN 978-3-030-61377-8.