

Universidade Federal do Pampa

Marcos Vinícius Treviso

Modelo para Trabalho de Conclusão de Curso

Alegrete

2015

Marcos Vinícius Treviso

Modelo para Trabalho de Conclusão de Curso

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Fábio Natanael Kepler

Alegrete

2015

Marcos Vinícius Treviso

Modelo para Trabalho de Conclusão de Curso

Projeto de Trabalho de Conclusão de Cursoa-
presentado ao Curso de Graduação em Ciên-
cia da Computação da Universidade Federal
do Pampa como requisito parcial para a ob-
tenção do título de Bacharel em Ciência da
Computação.

Projeto de Trabalho de Conclusão de Curso defendido e aprovado em de de
.....

Banca examinadora:

Fábio Natanael Kepler
Orientador

Professor
sigla da instituição

Professor
sigla da instituição

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

O esforço para chegar até aqui é grande, e certamente não conseguiria ter feito isso sozinho, por isso gostaria de agradecer meu orientador Fábio por me ajudar em assuntos variados durante a faculdade, com o qual aprendi muito.

Não é só de estudo que concluimos uma faculdade, depende muito da nossa força de vontade, e nesse quesito gostaria de agradecer muito a minha irmã Thaís, e especialmente meu pai Gilmar e minha mãe Loeiri por todos os conselhos e apoio.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

*"Existem muitas hipóteses em ciência que estão erradas. Isso é
perfeitamente aceitável, eles são a abertura para achar as que estão certas."
(Carl Sagan - O Mundo Assombrado Pelos Demônios)*

Resumo

Segundo a [ABNT \(2003, 3.1-3.2\)](#), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: Aprendizado de máquina. Processamento de linguagem natural. Part-of-speech tagging. Redes neurais. Word embeddings.

Abstract

This is the english abstract.

Key-words: Machine Learning. Natural language processing. Part-of-speech tagging. Neural networks. Word embeddings.

Lista de ilustrações

| | |
|---|----|
| Figura 1 – Exemplo de classificação gramatical | 27 |
| Figura 2 – Diagrama para resultado final | 31 |
| Figura 3 – Demonstração de regressão | 32 |
| Figura 4 – Demonstração de classificação | 33 |
| Figura 5 – Função sigmoide | 35 |
| Figura 6 – Representação dos casos da função de custo | 37 |
| Figura 7 – Função de custo - $J(\theta_0, \theta_1)$ | 38 |
| Figura 8 – Funcionamento do Gradiente Descendente | 40 |
| Figura 9 – Exemplo de <i>underfitting</i> e <i>overfitting</i> | 41 |
| Figura 10 – Exemplo de criação de <i>word embeddings</i> | 42 |
| Figura 11 – t-SNE: Visualização para <i>word embeddings</i> | 43 |
| Figura 12 – Exemplo de palavras similares | 44 |
| Figura 13 – Árvore de recursão de Fibonacci. | 50 |
| Figura 14 – Gráfico produzido diretamente no arquivo fonte. | 50 |
| Figura 15 – Outro gráfico feito em \LaTeX | 51 |
| Figura 16 – Variação dos resultados utilizando seleção por Janela Deslizante. | 52 |
| Figura 17 – Gráfico produzido em Excel e salvo como PDF. | 53 |
| Figura 18 – Exemplo de subfiguras. | 53 |
| Figura 19 – Mesmo exemplo de subfiguras, agora em escala. | 54 |
| Figura 20 – Exemplo de tamanhos de fonte | 73 |
| Figura 21 – Tela do JabRef para uma versão inicial do arquivo bib deste documento | 75 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Nomenclaturas | 34 |
| Tabela 2 – Níveis de investigação | 49 |
| Tabela 3 – Exemplo de tabela com dados de arquivo. | 49 |
| Tabela 4 – Linhas de código para inserir figura | 74 |

Lista de abreviaturas

Fig. Figura

Lista de siglas

PLN Processamento de Linguagem Natural

POS Part-of-speech

TCC Trabalho de Conclusão de Curso

Lista de símbolos

(x, y) Um exemplo de treinamento

$(X^{(i)}, Y^{(i)})$ O i -ésimo exemplo de treinamento

\mathcal{T} Conjunto de trabalhos de TCC

θ Parâmetros a serem aprendidos

A_i Área do i^{esimo} componente

m Número de exemplos de treinamento

X Dados de entrada ou *features*

$X^{(i)}$ O vetor coluna de todas as *features* no i -ésimo exemplo de treinamento

$X_j^{(i)}$ O valor da *feature* j no i -ésimo exemplo de treinamento

Y Dados de saída

Sumário

| | | |
|-------------|--|-----------|
| 1 | INTRODUÇÃO | 27 |
| 1.1 | Objetivo | 27 |
| 1.2 | Estrutura do trabalho | 28 |
| 2 | O PROBLEMA | 29 |
| 3 | FUNDAMENTOS | 31 |
| 3.1 | Aprendizado de máquina | 31 |
| 3.1.1 | Classificação | 33 |
| 3.1.2 | Função de hipótese | 34 |
| 3.1.3 | Função de custo | 36 |
| 3.1.4 | Minimização da função de custo | 38 |
| 3.1.5 | Classificação multiclasse | 40 |
| 3.1.6 | Regularização | 40 |
| 3.2 | Cópus e seu conjunto de classes gramaticais | 41 |
| 3.3 | Representação das palavras | 42 |
| 3.4 | Redes neurais | 43 |
| 3.5 | Aprendizagem profunda | 44 |
| 4 | TRABALHOS RELACIONADOS | 45 |
| 5 | DESENVOLVIMENTO | 47 |
| 5.1 | Formatação | 47 |
| 5.2 | Codificação dos arquivos: UTF8 | 47 |
| 5.3 | Citações | 47 |
| 5.3.1 | Referências internas | 48 |
| 5.4 | Tabelas | 48 |
| 5.5 | Figuras | 49 |
| 5.5.1 | Sobre a indicação da fonte de uma tabela ou figura | 51 |
| 5.6 | Expressões matemáticas | 53 |
| 5.7 | Enumerações: alíneas e subalíneas | 54 |
| 5.8 | Espaçamento entre parágrafos e linhas | 55 |
| 5.9 | Inclusão de outros arquivos | 56 |
| 5.10 | Compilar o documento \LaTeX | 56 |
| 6 | CONCLUSÃO | 59 |

| | |
|-----------------------|----|
| Referências | 61 |
|-----------------------|----|

APÊNDICES 63

| | |
|--|----|
| APÊNDICE A – DERIVADAS PARCIAIS DO GRADIENTE DESCENDENTE | 65 |
|--|----|

| | |
|------------------------|----|
| APÊNDICE B – | 67 |
|------------------------|----|

ANEXOS 69

| | |
|---|----|
| ANEXO A – FUNDAMENTOS DE ÁLGEBRA LINEAR | 71 |
|---|----|

| | |
|--|----|
| ANEXO B – LATEX PARA PRINCIPIANTES | 73 |
|--|----|

| | |
|----------------------------|----|
| B.1 Outras Dicas | 74 |
|----------------------------|----|

| | |
|------------------|----|
| Índice | 77 |
|------------------|----|

1 Introdução

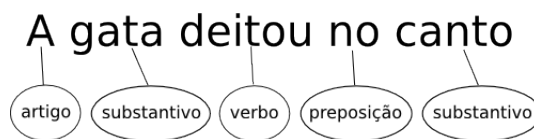
O ato de classificar uma palavra pertencente a um conjunto de textos em uma classe gramatical depende de sua estrutura morfo-sintática, esse ato é conhecido no campo de Processamento de Linguagem Natural (PLN) como Part-of-speech (POS) Tagging. A Figura 1 ilustra esse processo.

O conjunto de textos denominados *corpus* são amplamente utilizados para esse processo, e é sobre eles que é feito o treinamento do modelo de reconhecimento de padrões para que seja possível classificar uma palavra a uma certa classe gramatical.

Um dos problemas dessa classificação é justamente a eficiência com o qual cada classe gramatical é atribuída para certa palavra, nesse quesito, há vários métodos já idealizados que conseguiram uma eficiência de cerca de 97%, tais métodos são citados em (SANTOS; ZADROZNY, 2014; COLLOBERT, 2011; FONSECA; ROSA; ALUÍSIO, 2015). Fonseca, Rosa e Aluísio (2015) afirmam ter conseguido o estado-da-arte atual para o português com 97,42% de acurácia.

Apesar de muitos desses métodos já serem utilizados em larga escala, em PLN estamos sempre buscando ganhar mais performance, já que os POS Taggers podem ser aplicados em uma grande variedade de aplicações como tradução automática e extração de informações de textos (MANNING; SCHÜTZE, 1999), ferramentas de auxílio à leitura e escrita (MARQUIAFÁVEL, 2010), entre outras.

Figura 1: Exemplo de classificação gramatical



Nosso trabalho consiste em classificar palavras de acordo com seu contexto, ou seja, é feita a análise em termos das unidades primitivas que a compõem, e uma vez que ela é cumprida, podemos aplicar o resultado em outras análises.

1.1 Objetivo

Este trabalho tem, por fim, propor um novo método de classificação de palavras em classes gramaticais e analisar sua eficiência em relação a trabalhos já publicados que

utilizam métodos já consolidados. Primordialmente, isso será feito para o escopo da língua portuguesa brasileira.

Para buscar uma boa eficiência será proposto um método original, que se baseia em classificar primeiramente palavras mais fáceis (*e.g verbos*), desse modo, espera-se deixar palavras ambíguas por último. Por exemplo, para a [Figura 1](#) deixaríamos a palavra *canto* para o final, uma vez que não sabemos se isso se trata de um verbo ou de um substantivo. Além disso, o método proposto também terá como base a utilização de valores para cada classe gramatical de uma palavra, seguida por outras duas classes gramaticais das palavras anteriores.

Como já mencionado, o estado-da-arte atual tem atualmente cerca de 97% de acurácia, tentaremos ultrapassar esse limite aplicando novas técnicas de classificação e utilizando características significantes das palavras. É importante salientar que a acurácia da classificação não será a única medida levada em consideração, o tempo de processamento gasto no treinamento para cada *córpus* também será.

Para exemplificar brevemente, no final pretende-se que o modelo seja capaz de classificar corretamente as palavras analisando o seu contexto, e com isso outros analisadores conseguirão identificar o uso das palavras de acordo com seu significado, como mostrado abaixo na [Equação 1.1](#). Essa equação será explicada na [seção 3.3](#).

$$\text{homem} + \text{coroa} = \text{rei} \quad (1.1)$$

1.2 Estrutura do trabalho

A fim de proporcionar uma boa interpretação, este trabalho foi dividido nos seguintes capítulos: No [Capítulo 2](#) será descrito o problema de etiquetagem. No [Capítulo 3](#) serão mostrados os principais fundamentos necessários para entender o método proposto e seus conceitos relacionados; após passar os fundamentos, será apresentado no [Capítulo 4](#) os trabalhos relacionados que procuram resolver o problema de POS Tagging utilizando diferentes técnicas e abordagens. Depois, no [Capítulo 5](#) será detalhado aspectos da metodologia a ser aplicada e a explicação do método proposto juntamente com as técnicas utilizadas. Então, no ?? será feito um comparativo entre os resultados preliminares alcançados e uma breve discussão a respeito. E por fim, no [Capítulo 6](#) será apresentado as considerações finais.

2 O problema

Primeiramente, devemos informar que o problema a ser resolvido não é fácil, pois uma palavra pode possuir diferentes categorias gramaticais, e para solucioná-lo é preciso, na maioria das vezes, analisar todo o contexto no qual a palavra está associada. Além disso, como já mostrado anteriormente, há muita ambiguidade no português do Brasil, e essa ambiguidade dificulta a análise morfo-sintática, porque não é possível determinar a priori qual classe gramatical a palavra pertence. No exemplo anterior, da [Figura 1](#), não fica claro qual é o tipo de canto que está sendo referenciado, onde pode ser um verbo ou um substantivo (*o ato de cantar na primeira pessoa do singular do presente; um substantivo que se refere a um ângulo saliente; ou um substantivo que se refere a uma composição poética*). A ambiguidade pode muitas vezes ser resolvida se o contexto for analisado, como por exemplo:

"O rio da cidade."

A palavra *rio* pode ser classificada como substantivo ou como verbo, mas levando em consideração o sentido da frase, fica evidente que a classificação correta é substantivo.

Uma estratégia trivial seria utilizar um longo dicionário com uma função de mapeamento de um para um, onde a *chave* seria a palavra e o *valor* seria a classe gramatical. Infelizmente essa técnica é inviável com os recursos computacionais que temos atualmente, visto que o número de entradas seria estupidamente grande por ter todas as palavras possíveis de português, caso contrário, haveria o problema de ter uma palavra fora do vocabulário, e portanto ela não teria uma classe gramatical associada. Outro problema dessa estratégia é a ambiguidade, que faz com que uma palavra tenha mais que uma classe gramatical associada, e portanto não é possível mapear com indubitabilidade de que a classe associada é a correta sem antes de analisar o contexto.

Dito isso, o problema deste trabalho consiste em desenvolver um método para classificar palavras em suas respectivas classes gramaticais de modo eficiente. Para solucionar esse problema uma abordagem que está sendo amplamente utilizada é aprendizado de máquina, pois ela permite treinar um modelo que aprenda a classificar as palavras de acordo com o contexto em que está associada.

Em ordem de conseguir solucionar esse problema com eficiência, é necessário escolher um bom método computacional. Este trabalho se baseará na utilização de um método de aprendizado de máquina conhecido como redes neurais profundas (*deep-learning*), para que seja possível treinar modelos capazes de realizar a classificação. Utilizamos redes neu-

rais pois elas oferecem um jeito alternativo de realizar aprendizado de máquina quando temos hipóteses complexas com muitas características diferentes.

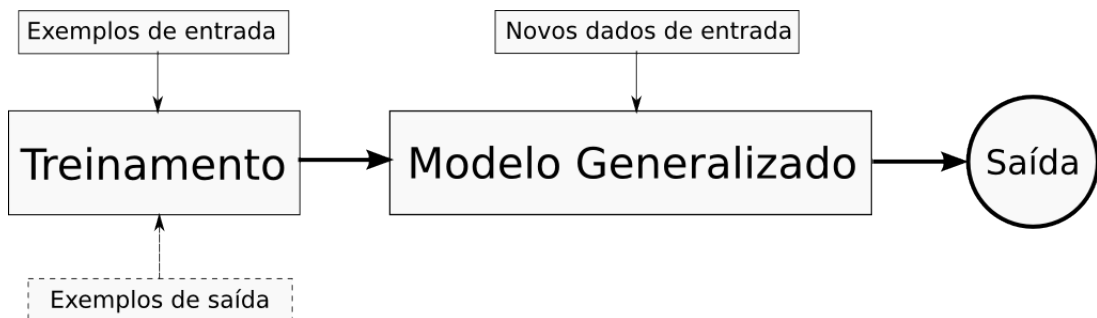
3 Fundamentos

Devido aos impasses demonstrados no [Capítulo 2](#), este trabalho é baseado em conceitos de aprendizado de máquina, que dedica-se na elaboração de algoritmos e técnicas que permitem um computador a aprender padrões.

3.1 Aprendizado de máquina

Para realizar a aprendizagem é necessário, a princípio, dados de entrada que servem como exemplo para nosso modelo, com esses dados é possível treinar o modelo para que ele possa então aprender com base nesses exemplos. Depois de realizar o treinamento, é possível generalizar sobre outros dados ainda não testados e gerar uma resposta apropriada como saída. O diagrama da [Figura 2](#) abaixo mostra os passos para obter o resultado final.

Figura 2: Diagrama para resultado final



O aprendizado de máquina pode ser dividido em duas abordagens: O **aprendizado supervisionado**, onde é dado um conjunto de dados de entrada e já se sabe como a saída deve parecer, tendo a ideia de que há uma relação entre a entrada e a saída; O **aprendizado não supervisionado**, que nos permite abordar problemas com pouca ou até nenhuma ideia de como os resultados devem parecer, nesse caso a caixa de exemplos de saída pode não existir.

Neste trabalho a abordagem utilizada vai ser o aprendizado supervisionado, pois já temos uma classe gramatical apropriada para cada palavra em seu contexto.

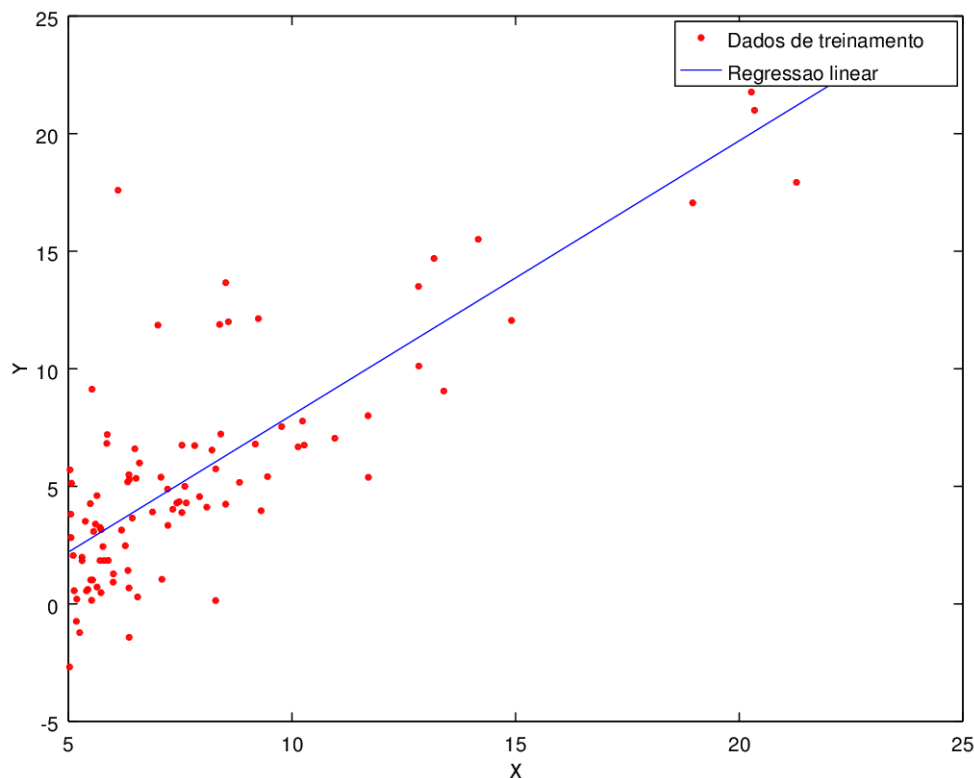
O aprendizado supervisionado permite dividir os problemas em duas categorias distintas:

- **Regressão:** Tenta-se prever resultados com uma saída contínua, significando que deseja-se mapear variáveis de entrada em alguma função contínua. A [Figura 3](#)

demonstra esse processo. Nessa demonstração, os círculos em vermelho são dados de exemplo, já a reta azul representa a predição feita pela regressão.

- **Classificação:** Tenta-se prever resultados em uma saída discreta, ou seja, deseja-se mapear variáveis de entrada em categoriais discretas. A [Figura 4](#) demonstra esse processo. Essa demonstração tem duas classes diferentes, as representadas pelos círculos amarelos e pelas cruzes vermelhas, já o contorno em verde representa o resultado da classificação sobre esses dados de exemplo.

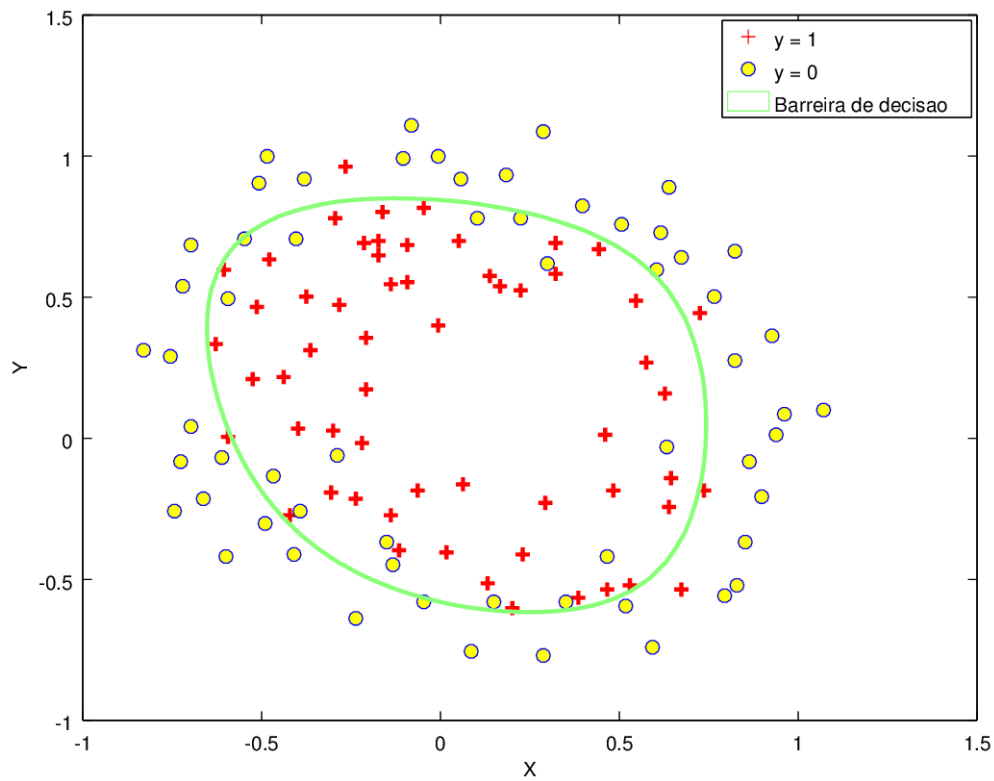
Figura 3: Demonstração de regressão



Fonte: [Ng \(2015\)](#)

Logo adiante será explicado os fundamentos de classificação e uma técnica para resolver esse problema chamada de regressão logística ([JR; LEMESHOW, 2004](#)). E em virtude de apresentar conceitos fundamentais há serem aplicados posteriormente nas redes neurais profundas, será falado um pouco mais sobre regressão logística, e também sobre propriedades importantes que serão referenciadas durante todo o trabalho.

Figura 4: Demonstração de classificação



Fonte: [Ng \(2015\)](#)

3.1.1 Classificação

Em um problema de classificação, nossa saída Y vai ser um vetor com valores sendo apenas zero ou um.

$$Y \in \{0, 1\} \quad (3.1)$$

A equação 3.1 está tratando apenas duas classes. Sendo assim, esse problema é chamado de classificação binária. Para resolver esse problema, um método que pode ser utilizado é regressão logística.

A fim de simplificar o uso das variáveis, faremos o uso de uma notação que é normalmente utilizada em textos de aprendizado de máquina, ela pode ser vista na [Tabela 1](#).

Tabela 1: Nomenclaturas

| | |
|----------------------|---|
| X | dados de entrada ou <i>features</i> |
| Y | dados de saída |
| $X_j^{(i)}$ | o valor da <i>feature</i> j no i -ésimo exemplo de treinamento |
| $X^{(i)}$ | o vetor coluna de todas as <i>features</i> no i -ésimo exemplo de treinamento |
| m | número de exemplos de treinamento |
| n | $ X^{(i)} $, o número de <i>features</i> |
| (x, y) | um exemplo de treinamento |
| $(X^{(i)}, Y^{(i)})$ | o i -ésimo exemplo de treinamento |
| θ | parâmetros a serem aprendidos |

Notação utilizada para classificação

3.1.2 Função de hipótese

A função de hipótese descreve a confiança dos dados de entrada para os dados de saída, para isso é atribuído valores para os parâmetros θ s. Essa relação está definida na [Equação 3.2](#). Observe que ela tem apenas dois parâmetros. Na realidade uma hipótese pode ter vários parâmetros relacionados com os dados de entrada X , se tiver n dados de entrada, então haverá $n+1$ parâmetros. A forma geral da função de hipótese está mostrada na [Equação 3.3](#).

$$h_{\theta}(x) = \theta_0 + \theta_1 x \quad (3.2)$$

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (3.3)$$

Uma maneira de simplificar a função de hipótese é trabalhar com a definição de multiplicação de matrizes. Então a função de hipótese pode ser representada de uma forma vetorizada, como mostrado na [Equação 3.4](#). Para maiores informações a respeito de matrizes e vetores, consultar o Anexo A, que traz informações básicas de álgebra linear.

$$h_{\theta}(x) = [\theta_0 \ \theta_1 \ \dots \ \theta_n] \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T X \quad (3.4)$$

Para que seja possível fazer operações de matrizes com θ e X , será setado $X_0^{(i)} = 1$ para todos os valores de i . Isso faz com que os vetores θ e $X^{(i)}$ combinem um com o outro elementarmente.

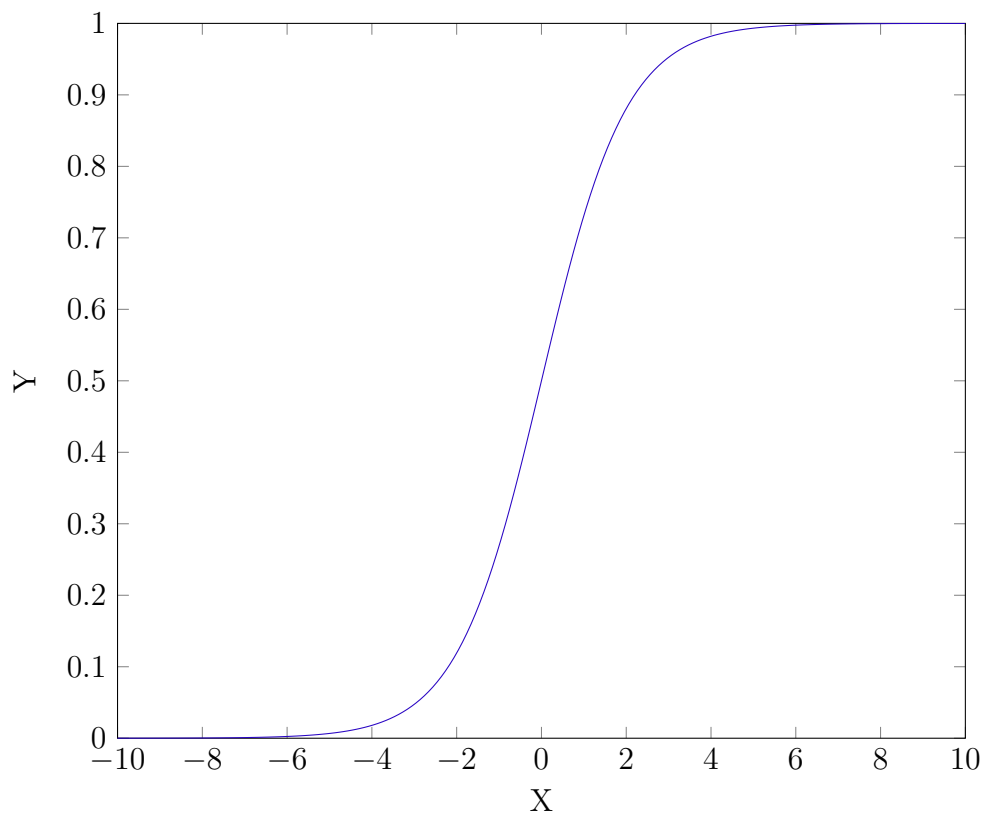
Uma função de hipótese tem o objetivo de mapear funções aleatórias dentro do intervalo de saída Y . Portanto, nossa função de hipótese deve satisfazer a [Equação 3.5](#).

$$0 \leq h_{\theta}(x) \leq 1 \quad (3.5)$$

Para fazer com que uma entrada contínua fique nesse intervalo, é necessário utilizar uma função linear que faça esse mapeamento de modo linear, e para isso mudamos a forma da função de hipótese. A nova forma usa a função sigmoide, também conhecida como função logística.

$$\begin{aligned} h_{\theta}(x) &= g(\theta^T X) \\ z &= \theta^T X \\ g(z) &= \frac{1}{1 + e^{-z}} \end{aligned} \quad (3.6)$$

Figura 5: Função sigmoide



A função da [Equação 3.6](#) representada na figura [Figura 5](#) mapeia qualquer número no intervalo $[0, 1]$, tornando isso útil para transformar qualquer valor arbitrário em uma função mais ideal para problemas de classificação.

Para calcular a função de hipótese, podemos colocar o $\theta^T X$ na função logística. Isso nos dará a probabilidade de que a saída é 1.

$$h_\theta(x) = P(y = 1|X; \theta) = 1 - P(y = 0|X; \theta)$$

Ou seja, a probabilidade de nossa previsão ser 1 é o oposto da probabilidade de ser 0. Sendo assim, a soma das probabilidades para $y = 0$ e $y = 1$ deve ser 1.

Para ter uma classificação discreta podemos traduzir a saída da função de hipótese de acordo com a [Equação 3.7](#):

$$\begin{aligned} h_\theta(x) \geq 0.5 &\Rightarrow y = 1 \\ h_\theta(x) < 0.5 &\Rightarrow y = 0 \end{aligned} \tag{3.7}$$

Então, se a entrada é $\theta^T X$, isso significa que:

$$h_\theta(x) = g(\theta^T X) \geq 0.5 \quad \text{quando} \quad \theta^T X \geq 0$$

A partir disso, podemos dizer que:

$$\begin{aligned} \theta^T X \geq 0 &\Rightarrow y = 1 \\ \theta^T X < 0 &\Rightarrow y = 0 \end{aligned}$$

O **limite de decisão** (ou barreira de decisão) é a linha que separa a área quando $y = 0$ e quando $y = 1$. Isso é criado pela função de hipótese. Uma observação importante é que a função de hipótese não precisa ser linear, pode ser até mesmo um círculo ou ter qualquer forma para ajustar os dados, como mostrado na [Figura 4](#). Isto é, uma função de hipótese pode ter várias *features*. Novas *features* podem ser criadas ao combinar as *features* já existentes e modificando-as. Por exemplo, se houvesse dois dados de entrada x_1 e x_2 , seria possível criar a função de hipótese:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_1^2 x_2^2 + \dots$$

Todavia, essa abordagem pode gerar problemas de desempenho. Algumas *features* podem também acabar se tornando obsoletas e não ajudar em nada no resultado final, tornando seu cálculo inútil. Mas o principal problema dessa abordagem é o *overfitting*, que faz com que a função de hipótese interpole os dados, esse problema será explicado na [subseção 3.1.6](#).

3.1.3 Função de custo

Para medir a precisão da função de hipótese é necessário uma função que diga o quão precisa aquela hipótese é, tal função é denominada de função de custo. Com ela, é

feita uma média de todos os resultados das hipóteses com a entrada X comparada com o valor objetivo Y . Ela pode ser expressa como na [Equação 3.8](#).

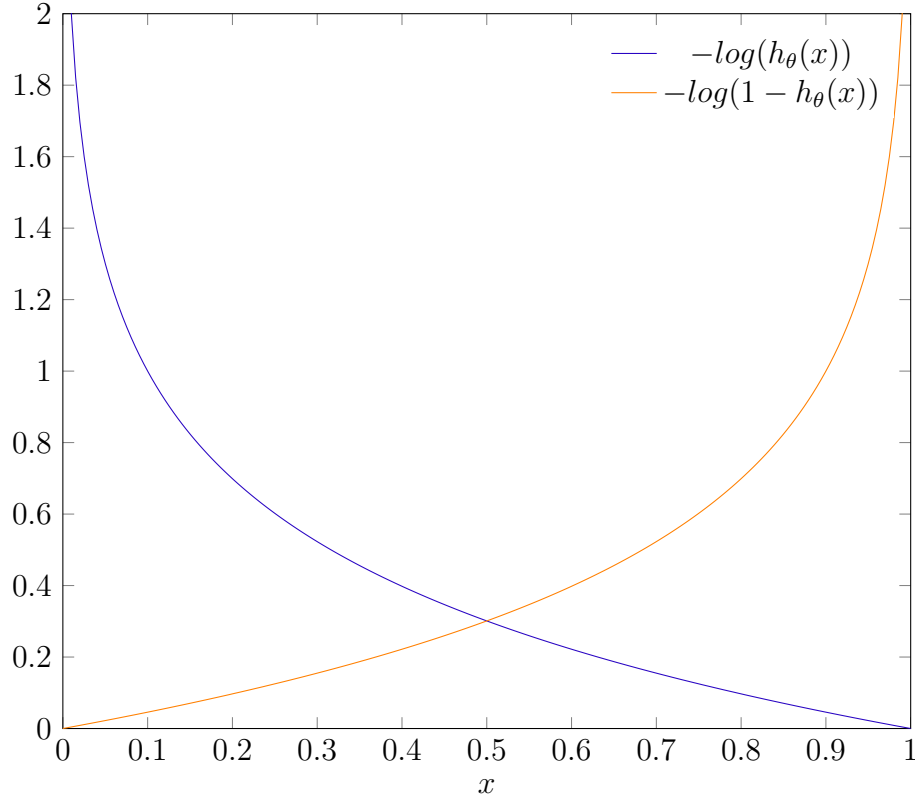
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Custo}(h_{\theta}(x^{(i)}, y^{(i)})) \quad (3.8)$$

Onde,

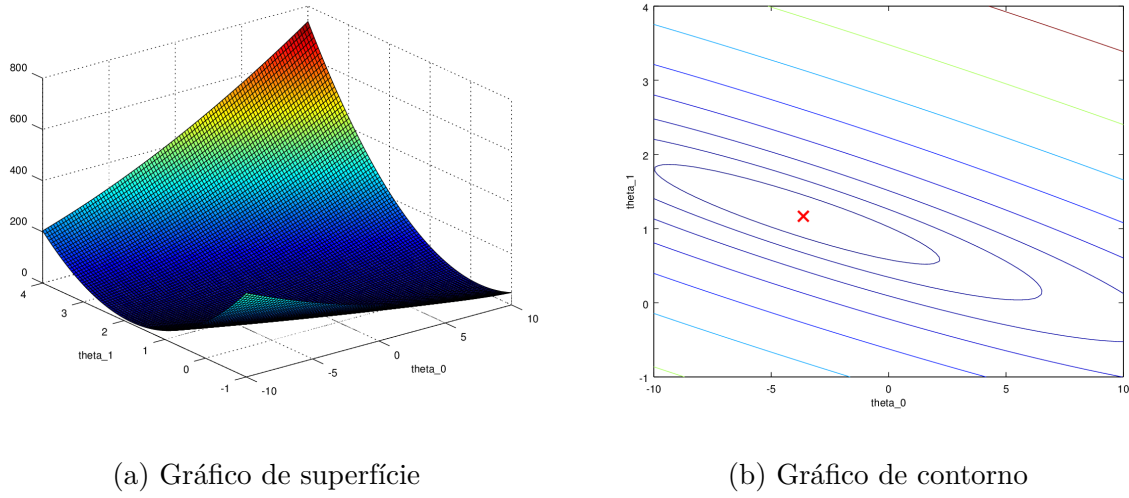
$$\text{Custo}(h_{\theta}(x^{(i)}, y^{(i)})) = \begin{cases} -\log(h_{\theta}(x)) & \text{se } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{se } y = 0 \end{cases} \quad (3.9)$$

A [Equação 3.9](#) captura a intuição de que se $h_{\theta}(x) = 0$, mas $y = 1$, então o algoritmo de aprendizado será penalizado por um custo muito alto. A [Figura 6](#) demonstra os dois respectivos casos dessa equação.

Figura 6: Representação dos casos da função de custo



O quanto mais a função de hipótese está longe de y , maior é a saída da função de custo. Se a função de hipótese é igual a y , então o custo é 0.

Figura 7: Função de custo - $J(\theta_0, \theta_1)$ 

Fonte: Ng (2015)

Podemos simplificar a [Equação 3.9](#) com dois casos condicionais em apenas um caso:

$$\text{Custo}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x)) \quad (3.10)$$

Nota-se que quando y é igual a 1, o segundo termo será 0 e não afetará o resultado. E quando y é igual a 0, o primeiro termo será 0 e não afetará o resultado. Aplicando a [Equação 3.10](#) na função de custo da equação [Equação 3.8](#), podemos reescrevê-la como a equação [Equação 3.11](#) ou na versão vetorizada da [Equação 3.12](#).

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right) \quad (3.11)$$

$$J(\theta) = -\frac{1}{m} \left(\log(g(X\theta))^T Y + \log(1 - g(X\theta))^T (1 - Y) \right) \quad (3.12)$$

O objetivo da regressão logística é minimizar a função de custo em relação aos parâmetros θ s. Usando dois parâmetros é possível visualizar uma função de custo através do número de iterações, isso pode ser observado no exemplo da [Figura ??](#) e da [Figura ??](#).

3.1.4 Minimização da função de custo

Para essa tarefa é necessário a aplicação de um algoritmo que pega a função de custo e tente minimizá-la, e por fim retorne os valores dos parâmetros θ s aprendidos. Com isso, estaremos melhorando nossa função de hipótese.

Um dos algoritmos mais utilizados para essa tarefa é o **Gradiente Descendente** (MICHALSKI; CARBONELL; MITCHELL, 2013), seu funcionamento baseia-se em seguir a derivada da função de custo em relação aos parâmetros θ s alternadamente, com isso a encosta da tangente dará a direção para seguir adiante. Ademais, é aplicada uma taxa de aprendizagem α a cada iteração, para tentar controlar a convergência.

Seu funcionamento é descrito pelo algoritmo 3.13.

repita até convergir {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (3.13)$$

}

Ao trabalhar sobre a derivada parcial da função de custo, podemos chegar na Equação 3.14 e adiante na versão vetorizada na Equação 3.15. A explicação de como trabalhar sobre a derivada pode ser encontrado no Apêndice A.

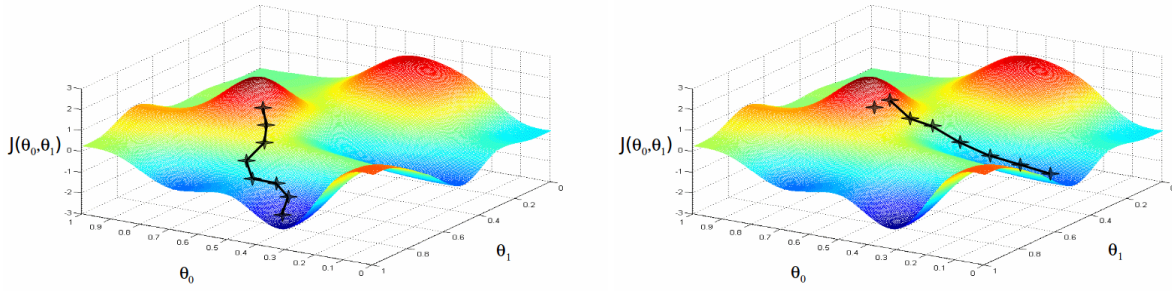
$$\theta_j := \theta_j - \alpha \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \quad (3.14)$$

$$\theta := \theta - \frac{\alpha}{m} X^T (g(X\theta) - Y) \quad (3.15)$$

Na Figura ?? é possível visualizar esse funcionamento sobre os parâmetros. Observe que nesse caso o algoritmo consegue convergir para o mínimo global, porém em alguns casos (dependendo de nossa taxa de aprendizagem α) pode-se convergir para um mínimo local como mostrado na Figura ?. Basicamente se α for um valor muito alto, os passos para o mínimo ótimo serão grandes e portanto o risco de divergência é maior, caso seja muito baixo, os passos para o mínimo ótimo serão pequenos e portanto vai demorar muito para convergir. Em (NG, 2015), é dito para testar manualmente esse valor, começando com 0,001 e subindo multiplicando por 10 esse valor até atingir uma divergência no resultado. Sendo assim possível analisar o melhor valor da taxa de aprendizagem e selecioná-la para o treinamento definitivo.

O Gradiente Descendente é apenas um dos vários algoritmos que existem para minimizar a função de custo. De acordo com (NG, 2015), há alternativas mais sofisticadas como o Gradiente Conjugado, BFGS e L-BFGS, onde além de serem mais rápidos que o Gradiente Descendente, eles não precisam selecionar manualmente o valor de α . Mas, também é sugerido que não devemos tentar codificar esses algoritmos, visto que são mais complexos e requerem um bom conhecimento de cálculo numérico.

Figura 8: Funcionamento do Gradiente Descendente



(a) Converging para uma mínimo global

(b) Converging para uma mínimo local

Fonte: Ng (2015)

3.1.5 Classificação multiclasse

O POS Tagging é um problema de classificação multiclasse, onde deve-se etiquetar uma palavra em uma de várias categorias gramaticais possíveis. É possível fazer isso ao expandir nossa definição para que a saída seja $y = \{0, 1, \dots, n\}$. Nesse caso, é dividido o problema em $n + 1$ problemas de classificação binária e em cada um será feito a predizagem da probabilidade de que y é membro de uma dessas classes.

$$h_{\theta}^{(0)}(X) = P(y = 0|X; \theta)$$

$$h_{\theta}^{(1)}(X) = P(y = 1|X; \theta)$$

$$\vdots$$

$$h_{\theta}^{(n)}(X) = P(y = n|X; \theta)$$

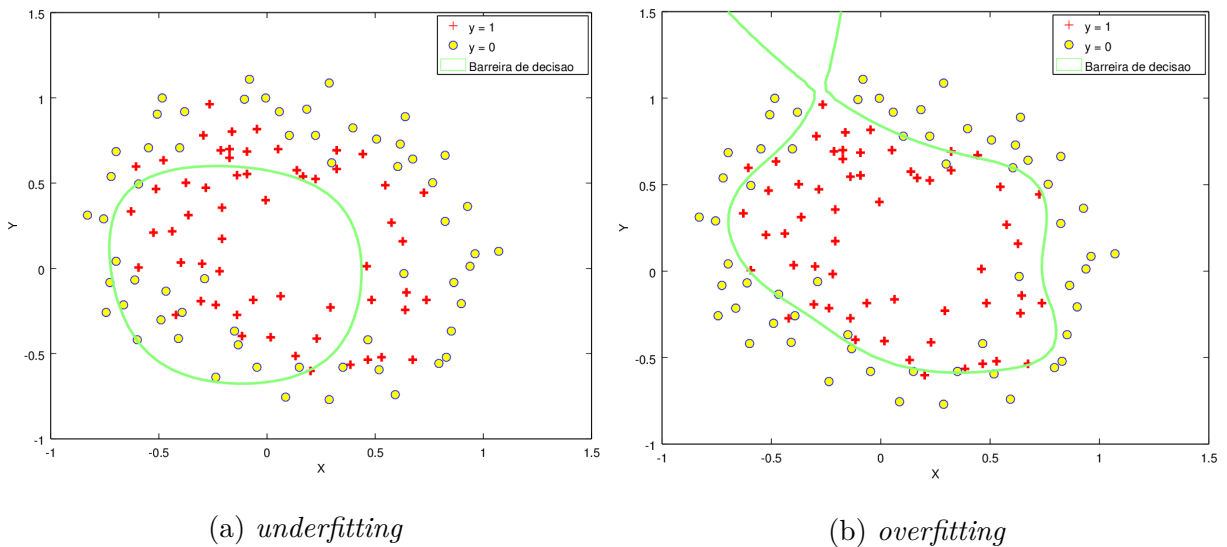
E então escolhe-se a classe com o valor de hipótese mais alto.

$$prediction = \max_i(h_{\theta}^{(i)}(X))$$

3.1.6 Regularização

Como já mencionado, regularização é um conceito importante designado para resolver o problema de *overfitting*.

Quando se trabalha com abordagens de aprendizagem supervisionada, tem dois problemas que podem ocorrer dependendo das *features* escolhidas. O ***underfitting*** ocorre quando a forma da função de hipótese mapeia mal à tendência dos dados. Isso é causado por uma função que é muito simples ou que usa pouquíssimas *features*. Em contrapartida, ***overfitting*** é causado por uma função de hipótese que encaixa os dados avaliados mas não generaliza bem para classificar novos dados. É usualmente causado por uma função complexa que cria muitas curvas e ângulos desnecessários não relacionados aos dados.

Figura 9: Exemplo de *underfitting* e *overfitting*

Fonte: Ng (2015)

A classifica33o ideal para o conjunto de dados mostrados na Figura 9 33 o exemplo mostrado na Figura 4.

Segundo (NG, 2015), h3 duas op33es principais para resolver o problema de *overfitting*:

- a) Reduzir o n3mero de *features*:
 - Manualmente selecionar quais *features* usar;
 - Usar um algoritmo de sele33o de modelo.
- b) Regulariza33o:
 - Manter todos as *features*, mas reduzir os par3metros θ_j .

Em geral, regulariza33o funciona bem quando h3 bastante *features* levemente 3teis.

3.2 C3rpus e seu conjunto de classes gramaticais

C3rpus s3o uma cole33o de textos colocados como um s3, eles s3o os exemplos de entrada para nosso treinamento. A qualidade e o tamanho desses c3rpus s3o importantes para que o etiquetador possa generalizar senten3as ainda n3o vistas. Eles cont3m tamb3m um conjunto de classes gramaticais associadas a cada palavra.

Esses conjuntos podem se diferenciar na sua granularidade, como por exemplo, podem ter diferentes classes gramaticais para nomes no plural e no singular, ou agrupar eles em uma 3nica classe (FONSECA; ROSA; ALU3SIO, 2015). No ingl3s, h3 v3rios

conjunto de classes gramaticais de qualidade que são amplamente utilizados, são o Penn Treebank tagset (THE..., 2014), CLAWS5 e CLAWS7.

Já no português, os corpus estão evoluindo com o tempo, embora alguns erros são encontrados neles (FONSECA; ROSA, 2013), eles ainda são a melhor opção devido a seu tamanho e qualidade, os principais e mais utilizados para essa tarefa são o Mac-Morpho (ALUÍSIO et al., 2003) com cerca de um milhão de palavras, que retrata artigos publicados na Folha de São Paulo em 1994. O Tycho Brahe (TEMPONI, 2004) que também tem cerca de um milhão de palavras e retrata assuntos históricos de 66 textos diferentes. Temos também o Bosque (AFONSO et al., 2002), que contém cerca de 185 mil palavras. Além desses, (FONSECA; ROSA; ALUÍSIO, 2015) apresenta uma versão revisada do Mac-Morpho, com classes gramaticais novas e junção de outras.

Infelizmente esses corpus não podem ser combinados em um só, já que eles se diferenciam no conjunto de classes gramaticais e também no seu uso associado. Uma possível alternativa para essa combinação, seria o uso de uma aprendizagem não supervisionada, aplicando técnicas de clusterização.

A aprendizagem baseada em corpus tem se mostrado uma estratégia atrativa, já que pode ser usado recursos criados com melhores performance [X, Y, Z]. Devido a isso, nesse trabalho vamos utilizar três corpus: o Mac-Morpho original; Mac-Morpho revisado e o Tycho Brahe. O conjunto de etiquetas para esses corpus podem ser encontrados, respectivamente, em (ALUÍSIO et al., 2003), (FONSECA; ROSA; ALUÍSIO, 2015) e (TEMPONI, 2004).

3.3 Representação das palavras

Palavras¹ podem ser representadas de várias maneiras em um modelo de aprendizagem, podendo ser até mesmo feito o uso real da palavra como um conjunto de caracteres.

No entanto, a melhor abordagem e a mais indicada atualmente é a utilização de *word embeddings*, elas são representação de palavras como vetores reais valorados em um espaço multidimensional (TURIAN; RATINOV; BENGIO, 2010). Elas podem ser geradas de maneiras diferentes dependendo da técnica utilizada, clássicas abordagens baseiam-se na frequência e co-ocorrência das palavras, até mesmo via modelos de redes neurais (SANTOS; ZADROZNY, 2014). Um dos processos de criação é mostrado na Figura 10.

Figura 10: Exemplo de criação de *word embeddings*

| Brasil | possui | vulcão | →

| | | |
|------|------|-----|
| 0.51 | 0.35 | ... |
|------|------|-----|

| | | |
|------|-------|-----|
| 0.66 | -0.21 | ... |
|------|-------|-----|

| | | |
|------|------|-----|
| 0.04 | 0.86 | ... |
|------|------|-----|

¹ Uma palavra pode ser qualquer conjunto de caracteres, inclusive pontuações, números, etc.

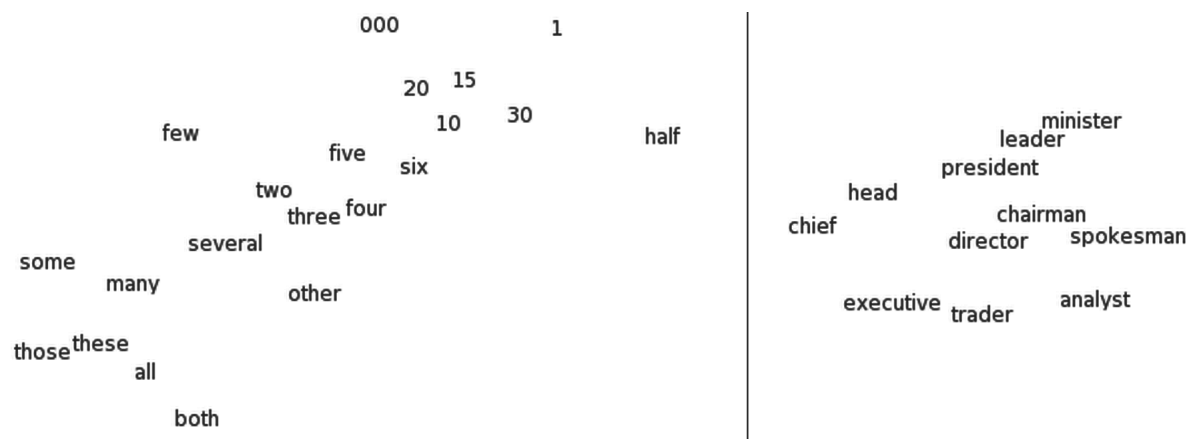
Ou seja, dada uma palavra *word*, temos a transformação W :

$$W : word \rightarrow \mathbb{R}^n \quad (3.16)$$

Em ordem de prever esses valores com precisão, a rede precisa aprender bons parâmetros para W .

A grande vantagem em se utilizar *word embeddings* é o fato dessa técnica conseguir capturar informações sintáticas e semânticas das palavras. Isso pode ser visualizado através do t-SNE (MAATEN; HINTON, 2008), uma técnica sofisticada para visualizar dados em altas dimensões.

Figura 11: t-SNE: Visualização para *word embeddings*



Fonte: [Turian, Ratinov e Bengio \(2010\)](#)

Na [Figura 11](#) podemos visualizar um tipo de mapeamento entre os sentidos intuitivos das palavras, onde na esquerda estão palavras referentes a números e na direita palavras referentes a profissões. Ou seja, palavras similares estão pertos umas das outras.

Na [Figura 12](#) temos mais alguns exemplos desse conceito, com isso podemos realizar tarefas semânticas mais facilmente, como o exemplo [1.1](#), que é o que esperamos obter.

Para mais informações sobre *word embeddings* consultar [X] e [Y].

3.4 Redes neurais

O que escrever aqui?

- Conceitos basicos de redes neurais

Figura 12: Exemplo de palavras similares

| FRANCE | JESUS | XBOX | REDDISH | SCRATCHED | MEGABITS |
|-------------|---------|-------------|-----------|-----------|------------|
| AUSTRIA | GOD | AMIGA | GREENISH | NAILED | OCTETS |
| BELGIUM | SATI | PLAYSTATION | BLUISH | SMASHED | MB/S |
| GERMANY | CHRIST | MSX | PINKISH | PUNCHED | BIT/S |
| ITALY | SATAN | IPOD | PURPLISH | POPPED | BAUD |
| GREECE | KALI | SEGA | BROWNISH | CRIMPED | CARATS |
| SWEDEN | INDRA | PSNUMBER | GREYISH | SCRAPED | KBIT/S |
| NORWAY | VISHNU | HD | GRAYISH | SCREWED | MEGAHERTZ |
| EUROPE | ANANDA | DREAMCAST | WHITISH | SECTIONED | MEGAPIXELS |
| HUNGARY | PARVATI | GEFORCE | SILVERY | SLASHED | GBIT/S |
| SWITZERLAND | GRACE | CAPCOM | YELLOWISH | RIPPED | AMPERES |

Fonte: [Collobert et al. \(2011\)](#)

3.5 Aprendizagem profunda

- Redes neurais profundas e seus algoritmos

4 Trabalhos relacionados

Vários métodos já foram propostos para resolver esse mesmo problema em português, apesar de nenhum deles ter um aproveitamento de 100% de precisão, muitos deles já são utilizados em escala global em diversas ferramentas formuladas por grandes empresas de tecnologia [X,Y].

É apresentado em [X] um etiquetador morfo-sintático baseado em cadeias de Markov de tamanho variável. Ele é testado sobre o corpus Tycho Brahe [X], ele apresenta uma precisão de 95.51%.

Em [Y] é apresentado um etiquetador que aprende automaticamente as features a serem usadas através de uma rede neural profunda que emprega uma camada evolutiva capaz de aprender a uma representação de caracteres das palavras. Isso foi testado sobre diferente corpus, no Mac-Morpho [X] foi alcançado uma precisão de 97.47%, no Tycho Brahe teve precisão de 97.17%, além disso, foi feito testes sobre uma versão revisada do Mac-Morpho distribuída por [Y] e obteve uma precisão de 97.17%

O mais recente etiquetador para o português brasileiro é mostrado em [X], onde é utilizado diferentes técnicas de representação das palavras e feita uma comparação entre elas. Nele é utilizado um modelo de rede neural profunda disponibilizado em [Y] e a melhor técnica obtém uma precisão de 97.47% no Mac-Morpho original.

5 Desenvolvimento

Alguns cuidados devem ser tomados no uso deste pacote. Leia as orientações a seguir e contate o responsável em caso de dúvidas.

5.1 Formatação

Embora não faça diferença no resultado final, é importante formatar adequadamente o seu código \LaTeX . Da mesma forma que para outras linguagens de programação, isso aumenta a legibilidade do código e ajuda a encontrar partes específicas mais rapidamente. As principais dicas para arquivos \TeX são:

- Indente seu código. Não só os ambientes (`begin`, `end`) mas também os parágrafos! Coloque cada sentença em uma linha, indentando a partir da segunda;
- Coloque marcações comentadas para delimitar o início de capítulos, seções, etc. Isso facilita buscar partes específicas em um arquivo.

Cuidado com abreviaturas e acrônimos. É fácil esquecer de os definir ou definir de maneira diferente em capítulos diferentes. Use os comandos do pacote `acro` para abreviaturas e acrônimos. Por exemplo, Figura ([Fig.](#)) é uma abreviação, então Trabalho de Conclusão de Curso ([TCC](#)) é um acrônimo/sigla. Eles são definidos no preâmbulo do documento.

Também vale a pena usar uma tabela de nomenclatura caso você use muitos símbolos, em especial símbolos matemáticos. Veja os comandos do pacote `nomenc1`. As definições também ficam no preâmbulo do documento.

5.2 Codificação dos arquivos: UTF8

A codificação de todos os arquivos deste pacote é **UTF8**. É necessário que você utilize a mesma codificação nos documentos que escrever, inclusive nos arquivos de bases bibliográficas `|.bib|`.

5.3 Citações

Utilize o ambiente `citacao` para incluir citações diretas com mais de três linhas:

As citações diretas, no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas. No caso de documentos datilografados, deve-se observar apenas o recuo (ABNT, 2002, 5.3)

Citações simples, com até três linhas, devem ser incluídas com aspas. Observe que em \LaTeX as aspas iniciais são diferentes das finais: “Amor é fogo que arde sem se ver”.

Para as citações indiretas, o comando padrão, `\cite`, realiza a forma mais comum de citação (`??`). A outra das formas mais usadas, para citar em texto corrido, é conseguida com o comando `\citeonline`: segundo `??`), na citação indireta, o número da página é opcional.

5.3.1 Referências internas

Usa-se o comando `\ref{}` para referenciar uma Tabela ou Figura. Por exemplo, esta é uma referência para a Tabela 2. Mas também pode-se usar o comando `\autoref{}`, que insere o tipo também. Por exemplo, esta é outra referência para a Tabela 2.

Há vários outros comandos interessantes. Eles estão no fonte do Capítulo 1, na subseção 5.3.1 ¹ (Introdução, página 27).

5.4 Tabelas

A Tabela 2 é um exemplo de tabela construída em \LaTeX . Como sugestão de formatação, evite ao máximo o uso de linhas verticais. As colunas de uma tabela devem ser separadas visivelmente. O contrário indica que a tabela está mal formatada ou que certas informações não deveriam estar nela.

Da mesma forma, evite o uso de linhas horizontais para separar linhas da tabela. Use-as apenas para separar o cabeçalho e eventuais partes importantes. Para obter um resultado ainda mais elegante, use os comandos do pacote `booktabs`.

Veja essas sugestões aplicadas na Tabela 2.

Uma opção avançada para a criação de tabelas é usar o pacote `pgfplotstable`. Ele permite que os dados de um arquivo sejam lidos e colocados em uma tabela, formatando-os da maneira que se quiser. A Tabela 3 é um exemplo. Veja o arquivo `desenvolvimento.tex` para os comandos necessários.

¹ O número do capítulo indicado é 1, que se inicia à página 27.

Tabela 2: Níveis de investigação.

| Nível de Inves- tigação | Insumos | Sistemas de Investigação | Produtos |
|----------------------------|--|-----------------------------|----------------------|
| Meta-nível | Filosofia da Ciência | Epistemologia | Paradigma |
| Nível do objeto | Paradigmas do metanível e evidências do nível inferior | Ciência | Teorias e modelos |
| Nível inferior | Modelos e métodos do nível do objeto e problemas do nível inferior | Prática | Solução de problemas |

Fonte: [van Gigch e Pipino \(1986\)](#)

Tabela 3: Exemplo de tabela com dados de arquivo.

| TAMANHO | MÉTODO 1 (%) | MÉTODO 2 (%) | MÉTODO 3 (%) | MÉDIA (%) |
|---------|--------------|--------------|--------------|-----------|
| 10 | 30.0 | 36.2 | 28.3 | 31.5 |
| 20 | 54.8 | 52.5 | 56.8 | 54.7 |
| 30 | 65.0 | 59.6 | 74.1 | 66.2 |
| 40 | 64.5 | 59.6 | 76.7 | 66.9 |
| 50 | 64.6 | 59.6 | 76.5 | 66.9 |

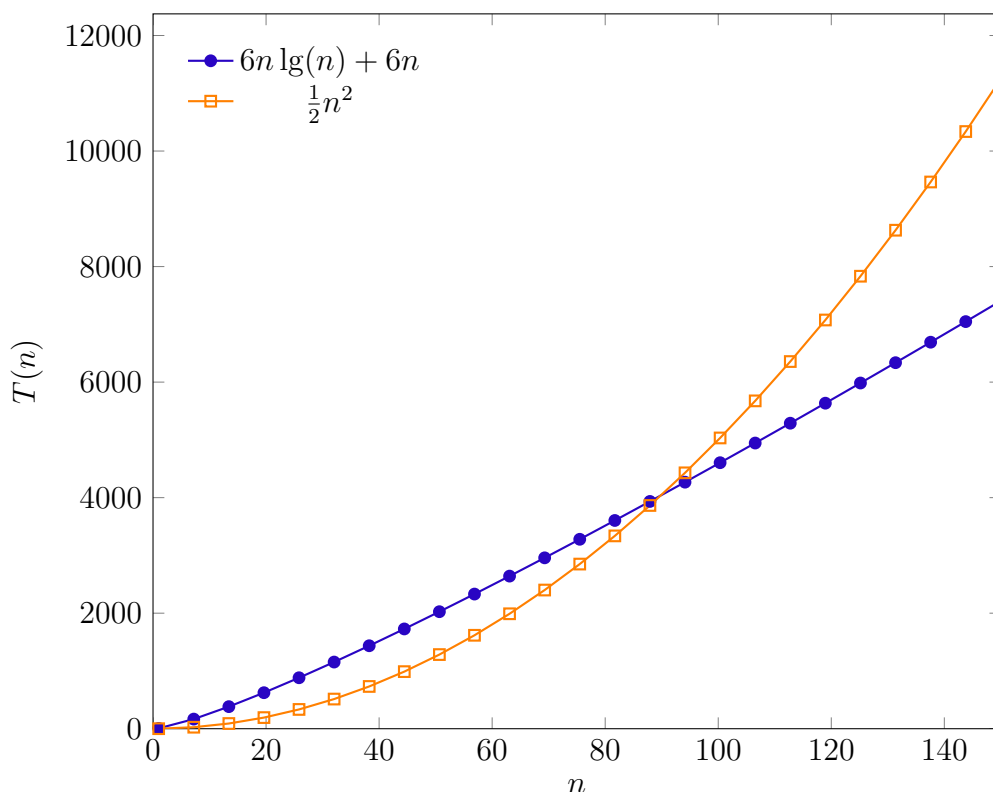
5.5 Figuras

Figuras podem ser criadas diretamente em \LaTeX . Uma das melhores formas, por ser relativamente simples, bem documentada e gerar ótimos resultados, é com o uso do pacote `tikz`². Ele permite gerar diagramas, árvores, fluxogramas etc. A [Figura 13](#) mostra um exemplo simples de árvore.

Junto com o pacote `pgfplots` também é possível gerar gráficos de funções ou a partir de dados em um arquivo (como no caso da [Tabela 3](#)). As [Figuras 14](#) e [15](#) mostram exemplos de gráficos de função, e a [Figura 16](#) um exemplo de gráfico a partir dos mesmos dados que os da [Tabela 3](#).

Figuras também podem ser incorporadas de arquivos externos, como é o caso da [Figura 17](#). Se a figura que ser incluída se tratar de um diagrama, um gráfico ou uma ilustração que você mesmo produza, priorize o uso de imagens vetoriais no formato PDF. Com isso, o tamanho do arquivo final do trabalho será menor, e as imagens terão uma apresentação melhor, principalmente quando impressas, uma vez que imagens vetoriais são perfeitamente escaláveis para qualquer dimensão. Nesse caso, se for utilizar o Microsoft Excel para produzir gráficos, ou o Microsoft Word para produzir ilustrações, exporte-os como PDF e os incorpore ao documento conforme o exemplo abaixo. No entanto, para manter a coerência no uso de software livre (já que você está usando \LaTeX e abnTeX2),

² Há vários exemplos em <http://www.texample.net/>.

Figura 15: Outro gráfico feito em L^AT_EX.

aprimorar as imagens incorporadas com o software livre **Gimp**⁴. Ele é uma alternativa livre ao Adobe Photoshop.

A Figura 18 na página 53 contém duas subfiguras, Figura 18(a) e (b). A Figura 18(a) foi inserida de um arquivo externo, enquanto a Figura 18(b) foi escrita dentro do próprio código T_EX. A Figura 19 contém o mesmo exemplo, mas usando comandos diferentes para inserir as Subfiguras 19(a) e (b).

Na Figura 18, as legendas (que indicam a fonte) para cada subfigura só funcionaram porque as figuras ficaram uma embaixo da outra. Se elas estivessem lado a lado, a inserção do comando `\legend` em cada uma faria com que elas ficassem organizadas na vertical. Uma legenda geral funcionaria, entretanto.

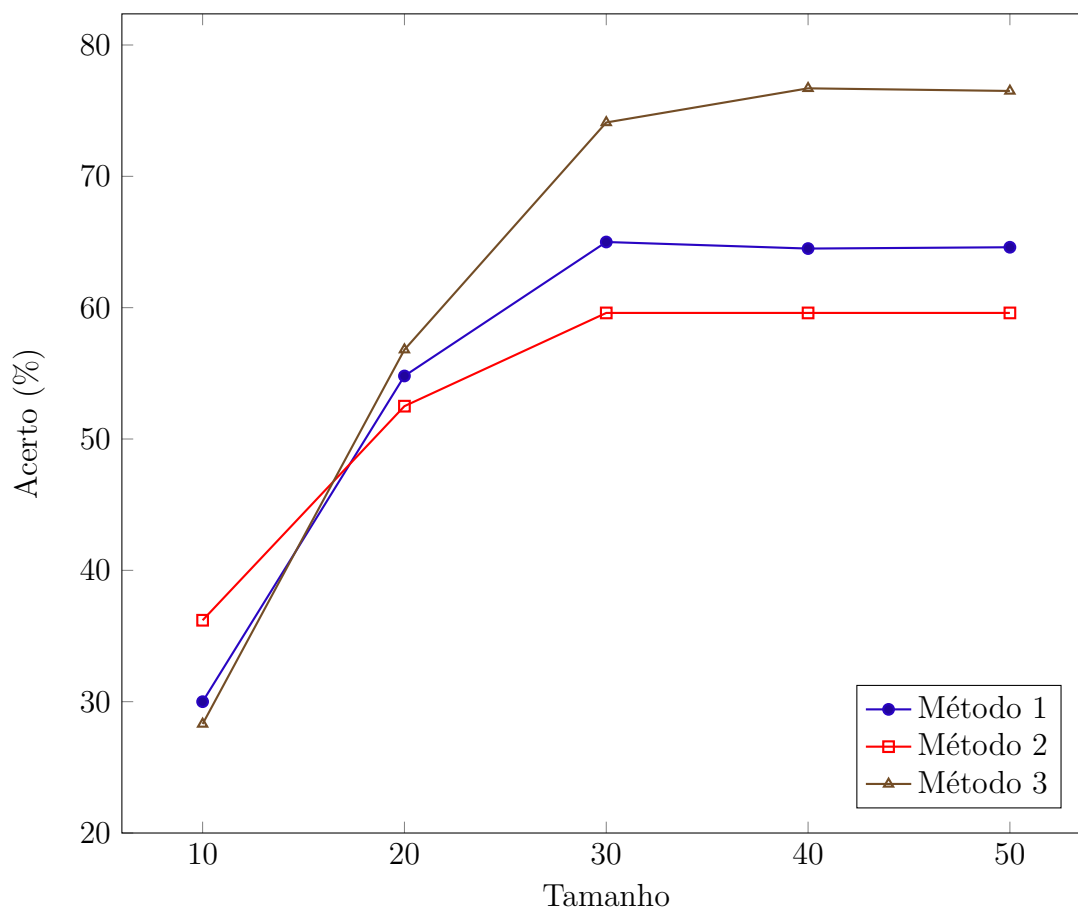
Na Figura 19, tanto legendas para subfiguras quanto uma legenda geral funcionam.

5.5.1 Sobre a indicação da fonte de uma tabela ou figura

As normas ABNT (2011, 5.8) e o Manual de Normatização da UNIPAMPA (??) dizem para, “Após a ilustração, na parte inferior, indicar a fonte consultada (elemento obrigatório, mesmo que seja produção do próprio autor), legenda, notas e outras infor-

⁴ <<http://www.gimp.org/>>

Figura 16: Variação dos resultados utilizando seleção por Janela Deslizante.

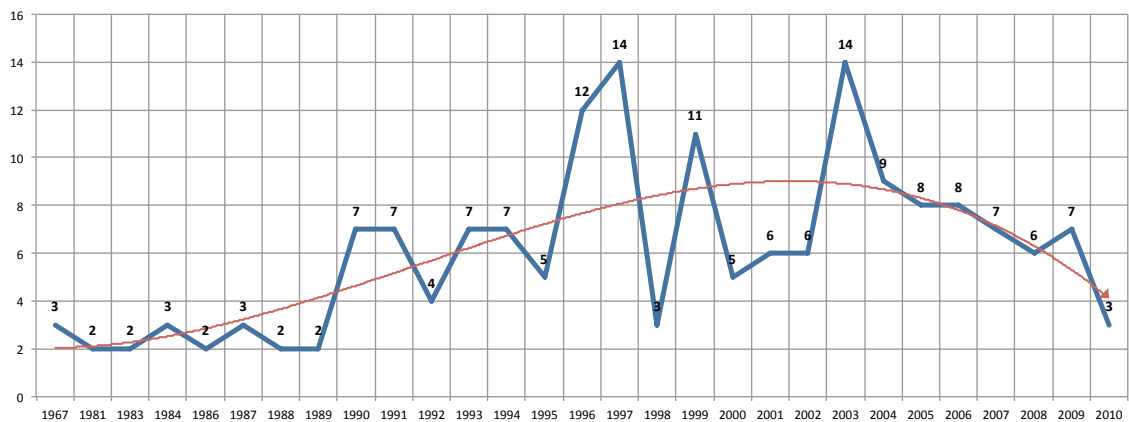


mações necessárias à sua compreensão (se houver).” A primeira interpretação é a de que, mesmo que o autor tenha criado a figura, a fonte deverá ser indicada. Com efeito, várias outras normas, manuais e inclusive o exemplo do pacote `abnTeX22` usam “Fonte: os autores” em alguns lugares.

Entretanto, isso não está correto. Veja o trecho em destaque: “Após a ilustração, na parte inferior, indicar a fonte **consultada** (elemento obrigatório, mesmo que seja produção do próprio autor) (...).” A interpretação correta é a de que, caso a ilustração tenha sido **extraída** de um documento, a fonte deve ser indicada, ainda que esse documento pertença ao próprio autor. A sentença original das normas deveria ter sido melhor escrita para evitar a interpretação incorreta.

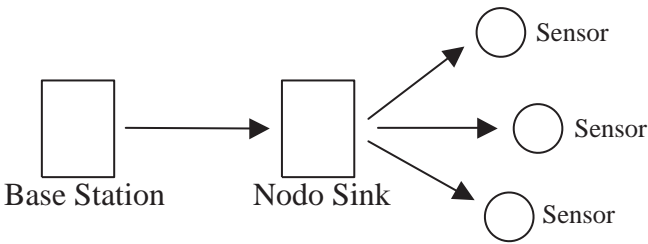
Assim, não indique a fonte se a figura ou tabela foi criada para o trabalho, ou seja, se é inédita. Caso contrário, indique a fonte. Mas cuidado: caso a figura ou tabela tenha sido adaptada de outra já publicada, então é obrigatório indicar “adaptado de” ou “acrescida de” seguido da referência da fonte de onde ela foi extraída.

Figura 17: Gráfico produzido em Excel e salvo como PDF.



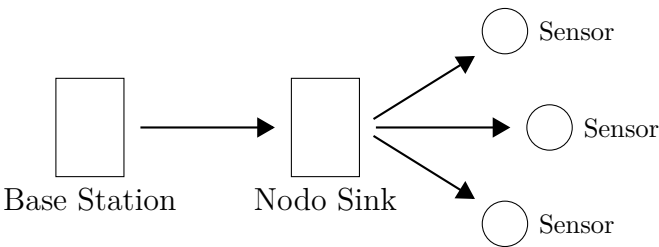
Fonte: [Araujo \(2012, p. 24\)](#)

Figura 18: Exemplo de subfiguras.



(a) Uma figura de um arquivo.

Fonte: ??)



(b) Uma figura em puro código TikZ.

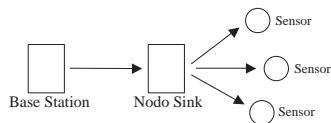
Alterado: de ??)

5.6 Expressões matemáticas

Use o ambiente `equation` para escrever expressões matemáticas numeradas:

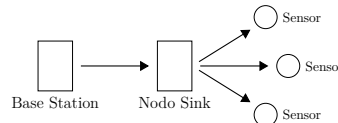
$$\forall x \in X, \quad \exists y \leq \epsilon \tag{5.1}$$

Figura 19: Mesmo exemplo de subfiguras, agora em escala.



((a)) Uma figura de um arquivo.

Fonte: ??)



((b)) Uma figura em puro código TikZ.

Alterado: de ??)

Fonte geral

Escreva expressões matemáticas entre \$ e \$, como em $\lim_{x \rightarrow \infty} \exp(-x) = 0$, para que fiquem na mesma linha.

Também é possível usar colchetes para indicar o início de uma expressão matemática que não é numerada.

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left(\sum_{i=1}^n a_i^2 \right)^{1/2} \left(\sum_{i=1}^n b_i^2 \right)^{1/2}$$

Consulte mais informações sobre expressões matemáticas em <http://code.google.com/p/abntex2/w/edit/Referencias>.

5.7 Enumerações: alíneas e subalíneas

Quando for necessário enumerar os diversos assuntos de uma seção que não possua título, esta deve ser subdividida em alíneas (ABNT, 2012, 4.2):

- a) os diversos assuntos que não possuam título próprio, dentro de uma mesma seção, devem ser subdivididos em alíneas⁵;
- b) o texto que antecede as alíneas termina em dois pontos;
- c) as alíneas devem ser indicadas alfabeticamente, em letra minúscula, seguida de parêntese. Utilizam-se letras dobradas, quando esgotadas as letras do alfabeto;
- d) as letras indicativas das alíneas devem apresentar recuo em relação à margem esquerda;
- e) o texto da alínea deve começar por letra minúscula e terminar em ponto-e-vírgula, exceto a última alínea que termina em ponto final;

⁵ As notas devem ser digitadas ou datilografadas dentro das margens, ficando separadas do texto por um espaço simples de entre as linhas e por filete de 5 cm, a partir da margem esquerda. Devem ser alinhadas, a partir da segunda linha da mesma nota, abaixo da primeira letra da primeira palavra, de forma a destacar o expoente, sem espaço entre elas e com fonte menor. ABNT (2011, 5.2.1)

- f) o texto da alínea deve terminar em dois pontos, se houver subalínea;
- g) a segunda e as seguintes linhas do texto da alínea começa sob a primeira letra do texto da própria alínea;
- h) subalíneas (ABNT, 2012, 4.3) devem ser conforme as alíneas a seguir:
 - as subalíneas devem começar por travessão seguido de espaço;
 - as subalíneas devem apresentar recuo em relação à alínea;
 - o texto da subalínea deve começar por letra minúscula e terminar em ponto-e-vírgula. A última subalínea deve terminar em ponto final, se não houver alínea subsequente;
 - a segunda e as seguintes linhas do texto da subalínea começam sob a primeira letra do texto da própria subalínea.
- i) no `abnTeX2` estão disponíveis os ambientes `incisos` e `subalineas`, que em suma são o mesmo que se criar outro nível de `alineas`, como nos exemplos à seguir:
 - *Um novo inciso em itálico;*
- j) Alínea em **negrito**:
 - *Uma subalínea em itálico;*
 - *Uma subalínea em itálico e sublinhado;*
- k) Última alínea com *ênfase*.

5.8 Espaçamento entre parágrafos e linhas

O tamanho do parágrafo, espaço entre a margem e o início da frase do parágrafo, é definido por:

```
\setlength{\parindent}{1.3cm}
```

Por padrão, não há espaçamento no primeiro parágrafo de cada início de divisão do documento (??). Porém, você pode definir que o primeiro parágrafo também seja indentado, como é o caso deste documento. Para isso, apenas inclua o pacote `indentfirst` no preâmbulo do documento:

```
\usepackage{indentfirst}      % Indenta o primeiro parágrafo de cada seção.
```

O espaçamento entre um parágrafo e outro pode ser controlado por meio do comando:

```
\setlength{\parskip}{0.2cm} % tente também \onelineskip
```

O controle do espaçamento entre linhas é definido por:

```
\OnehalfSpacing      % espaçamento um e meio (padrão);
\DoubleSpacing        % espaçamento duplo
\SingleSpacing         % espaçamento simples
```

Para isso, também estão disponíveis os ambientes:

```
\begin{SingleSpace} ... \end{SingleSpace}
\begin{Spacing}{hfactori} ... \end{Spacing}
\begin{OnehalfSpace} ... \end{OnehalfSpace}
\begin{OnehalfSpace*} ... \end{OnehalfSpace*}
\begin{DoubleSpace} ... \end{DoubleSpace}
\begin{DoubleSpace*} ... \end{DoubleSpace*}
```

Para mais informações, consulte [Wilson e Madsen \(2010, p. 47-52 e 135\)](#).

5.9 Inclusão de outros arquivos

É uma boa prática dividir o seu documento em diversos arquivos, e não apenas escrever tudo em um único. Esse recurso foi utilizado neste documento. Para incluir diferentes arquivos em um arquivo principal, de modo que cada arquivo incluído fique em uma página diferente, utilize o comando:

```
\include{documento-a-ser-incluido} % sem a extensão .tex
```

Para incluir documentos sem quebra de páginas, utilize:

```
\input{documento-a-ser-incluido} % sem a extensão .tex
```

5.10 Compilar o documento L^AT_EX

Geralmente os editores L^AT_EX, como o TeXlipse⁶, o Texmaker⁷, entre outros, compilam os documentos automaticamente, de modo que você não precisa se preocupar com isso.

⁶ <<http://texlipse.sourceforge.net/>>

⁷ <<http://www.xmlmath.net/texmaker/>>

No entanto, você pode compilar os documentos \LaTeX usando os seguintes comandos, que devem ser digitados no *Prompt de Comandos* do Windows ou no *Terminal* do Mac ou do Linux:

```
pdflatex ARQUIVO_PRINCIPAL.tex
bibtex ARQUIVO_PRINCIPAL.aux
makeindex ARQUIVO_PRINCIPAL.idx
makeindex ARQUIVO_PRINCIPAL.nlo -s nomencl.ist -o ARQUIVO_PRINCIPAL.nls
pdflatex ARQUIVO_PRINCIPAL.tex
pdflatex ARQUIVO_PRINCIPAL.tex
```


6 Conclusão

Bom trabalho!

Referências

- AFONSO, S. et al. Floresta sintá (c) tica: A treebank for portuguese. In: *LREC*. [S.l.: s.n.], 2002. Citado na página 42.
- ALUÍSIO, S. et al. An account of the challenge of tagging a reference corpus for brazilian portuguese. In: *Computational Processing of the Portuguese Language*. [S.l.]: Springer, 2003. p. 110–117. Citado na página 42.
- ARAUJO, L. C. *Configuração: uma perspectiva de Arquitetura da Informação da Escola de Brasília*. Dissertação (Mestrado) — Universidade de Brasília, Brasília, Março 2012. Citado na página 53.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 10520*: Informação e documentação — apresentação de citações em documentos. Rio de Janeiro, 2002. 7 p. Citado na página 48.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6028*: Resumo - apresentação. Rio de Janeiro, 2003. 2 p. Citado na página 11.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 14724*: Informação e documentação — trabalhos acadêmicos — apresentação. Rio de Janeiro, 2011. 15 p. Citado 2 vezes nas páginas 51 e 54.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6024*: Numeração progressiva das seções de um documento. Rio de Janeiro, 2012. 4 p. Citado 2 vezes nas páginas 54 e 55.
- COLLOBERT, R. Deep learning for efficient discriminative parsing. In: *International Conference on Artificial Intelligence and Statistics*. [S.l.: s.n.], 2011. Citado na página 27.
- COLLOBERT, R. et al. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, JMLR. org, v. 12, p. 2493–2537, 2011. Citado na página 44.
- FONSECA, E. R.; ROSA, J. L. G. Mac-morpho revisited: Towards robust part-of-speech tagging. In: *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*. [S.l.: s.n.], 2013. p. 98–107. Citado na página 42.
- FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, Springer, v. 21, n. 1, p. 1–14, 2015. Citado 3 vezes nas páginas 27, 41 e 42.
- JR, D. W. H.; LEMESHOW, S. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2004. Citado na página 32.
- MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, v. 9, n. 2579-2605, p. 85, 2008. Citado na página 43.

MANNING, C. D.; SCHÜTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999. Citado na página 27.

MARQUIAFÁVEL, V. S. Um processo para a geração de recursos lingüísticos aplicáveis em ferramentas de auxílio à escrita científica. Biblioteca Digital de Teses e Dissertações da Universidade Federal de São Carlos, 2010. Citado na página 27.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. *Machine learning: An artificial intelligence approach*. [S.l.]: Springer Science & Business Media, 2013. Citado na página 39.

NG, A. *Course of Machine Learning*. [S.l.], 2015. Disponível em: <<https://www.coursera.org/learn/machine-learning/>>. Citado 6 vezes nas páginas 32, 33, 38, 39, 40 e 41.

SANTOS, C. N. dos; ZADROZNY, B. Training state-of-the-art portuguese pos taggers without handcrafted features. In: *Computational Processing of the Portuguese Language*. [S.l.]: Springer, 2014. p. 82–93. Citado 2 vezes nas páginas 27 e 42.

TEMPONI, C. N. O corpus anotado do português histórico: um avanço para as pesquisas em lingüística histórica do português. *Revista Virtual de Estudos da Linguagem: ReVEL*, v. 2, n. 3, p. 1, 2004. Citado na página 42.

THE Penn Treebank Project (2014). [S.l.], 2014. Disponível em: <<http://www.cis.upenn.edu/~treebank/>>. Citado na página 42.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 48th annual meeting of the association for computational linguistics*. [S.l.], 2010. p. 384–394. Citado 2 vezes nas páginas 42 e 43.

van GIGCH, J. P.; PIPINO, L. L. In search for a paradigm for the discipline of information systems. *Future Computing Systems*, v. 1, n. 1, p. 71–97, 1986. Citado na página 49.

WILSON, P.; MADSEN, L. *The Memoir Class for Configurable Typesetting - User Guide*. Normandy Park, WA, 2010. Disponível em: <<http://ctan.tche.br/macros/latex/contrib/memoir/memman.pdf>>. Acesso em: 19.12.2012. Citado na página 56.

Apêndices

APÊNDICE A – Derivadas parciais do Gradiente Descendente

De acordo com a ABNT:

Apêndice (opcional): texto utilizado quando o autor pretende complementar sua argumentação. São identificados por letras maiúsculas e travessão, seguido do título. Ex.: APÊNDICE A - Avaliação de células totais aos quatro dias de evolução

Anexo (opcional): texto ou documento **não elaborado pelo autor** para comprovar ou ilustrar. São identificados por letras maiúsculas e travessão, seguido do título. Ex.: ANEXO A - Representação gráfica de contagem de células

Tais definições (e outras) podem ser encontradas na NBR 14724-2001 Informação e documentação - trabalhos acadêmicos¹.

¹ <http://www.firb.br/abntmonograf.htm>

APÊNDICE B –

Pode ser que tenha outro...

Anexos

ANEXO A – Fundamentos de Álgebra Linear

Sendo anexo, a formatação dessa seção é livre. Ou seja: aceita-se fonte diferente e menor

ANEXO B – LaTeX para Principiantes

TEste¹ Dentro dos arquivos .tex o texto pode estar organizado em partes, capítulos, seções, etc. conforme os seguintes comandos:

- `\part{Nome da Parte}`, partes do documento
- `\chapter{Nome}`, capítulos somente para arquivos do tipo *book* e *report*
- `\section{Nome}`, seções
- `\subsection{Nome}`, subseções
- `\subsubsection{Nome}`, seções dentro de subseções
- `\paragraph{Texto}`, parágrafos formatados
- `\subparagraph{Texto}`, subparágrafos

Parágrafos Parágrafos são definidos deixando uma linha em branco entre os mesmos. Pode-se também forçar usando `\\` bem como deixar uma linha em branco com um `~` sozinho na linha.

Formato de texto O tamanho do texto pode ser definido pelos comandos específicos: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `huge` e `Huge`, conforme ilustra a Figura 20.

| Command | Sample | Command | Sample |
|----------------------------|---------------------------|--------------------------|--------------------------|
| <code>\tiny</code> | <code>tiny</code> | <code>\scriptsize</code> | <code>scriptsize</code> |
| <code>\footnotesize</code> | <code>footnotesize</code> | <code>\small</code> | <code>small</code> |
| <code>\normalsize</code> | <code>normalsize</code> | <code>\large</code> | <code>large</code> |
| <code>\Large</code> | <code>larger</code> | <code>\LARGE</code> | <code>even larger</code> |
| <code>\huge</code> | <code>huge</code> | <code>\Huge</code> | <code>largest</code> |

Figura 20: Exemplo de tamanhos de fonte

Referências dentro do Texto Partes do texto podem ser referenciadas através do par de comandos `\label` e `\ref`. Por exemplo, podemos inserir uma seção no artigo utilizando o seguinte comando:

```
\section{Seção principal}\label{sec:prcpal}
```

Vejam que o título da seção é seguido do comando `\label{nome}`. Esta seção pode ser referenciada em qualquer parte do texto, como o exemplo a seguir.

Conforme explicado na Seção `\ref{sec:prcpal}`, nosso método utiliza...

¹ (??)

Perceba que a figura tem resolução ruim; deveria ser uma tabela

B.1 Outras Dicas

Caracteres Especiais Esses não podem ser usados no texto sem a barra à frente: # \$ % ^ & _ { } ~ e /.

Comentários Comentários são precedidos de % e podem estar em qualquer parte do texto. Lembrando que tudo que estiver após % será considerado como comentário e ignorado pelo processador.

Incluir Figuras Incluir figuras no LaTeX é relativamente fácil quando se tem um formato de arquivo pré-definido. Por exemplo, neste documento, usa-se apenas figuras do tipo *pdf*, mas também poderia-se usar do tipo *png* (e *jpeg*, mas este tipo não é recomendado). A Tabela 4 ilustra as linhas que inserem uma figura no texto.

Tabela 4: Linhas de código para inserir figura

| Linha de Código | Explicação |
|---|---|
| <code>\usepackage{graphicx}</code> | <i>inclui pacote gráfico no início do documento</i> |
| <code>\begin{figure}[tb]</code> | <i>inicia figura, define sua posição no texto</i> |
| <code>\centering</code> | <i>centraliza a figura na página</i> |
| <code>\includegraphics[scale=.7]</code> | <i>define escala da figura</i> |
| <code>{img/figura}</code> | <i>inclui o arquivo da figura no texto</i> |
| <code>\caption{Legenda}</code> | <i>inclui a legenda da figura</i> |
| <code>\label{fig:ap}</code> | <i>inclui o apelido da figura</i> |
| <code>\end{figure}</code> | <i>termina figura</i> |

Hifenização Às vezes aparece uma palavra cuja hifenização, divisão silábica, está errada. Para resolver esse tipo de problema, pode-se recorrer à divisão manual da palavra, acrescentando \- entre cada sílaba: `Mi\re\lla`. Se, ao invés desta solução, você quiser evitar completamente que suas palavras sejam divididas, acrescente os dois comandos no início do seu documento (ou seja, antes do `begin{document}`).

```
\hyphenpenalty=5000
\tolerance=1000
```

BibTeX Para editar facilmente o BibTeX, pode-se utilizar uma ferramenta própria². A minha favorita é o JabRef³, ilustrado na Figure 21, porque:

- É de graça;
- Possui interface gráfica super intuitiva;
- Permite importar referências de bases clássicas, como ISI, Medline e RIS;

² Ferramentas para BibTeX: <http://dmoz.org/Computers/Software/Typesetting/TeX/BibTeX>

³ JabRef Editor: <http://jabref.sourceforge.net/>

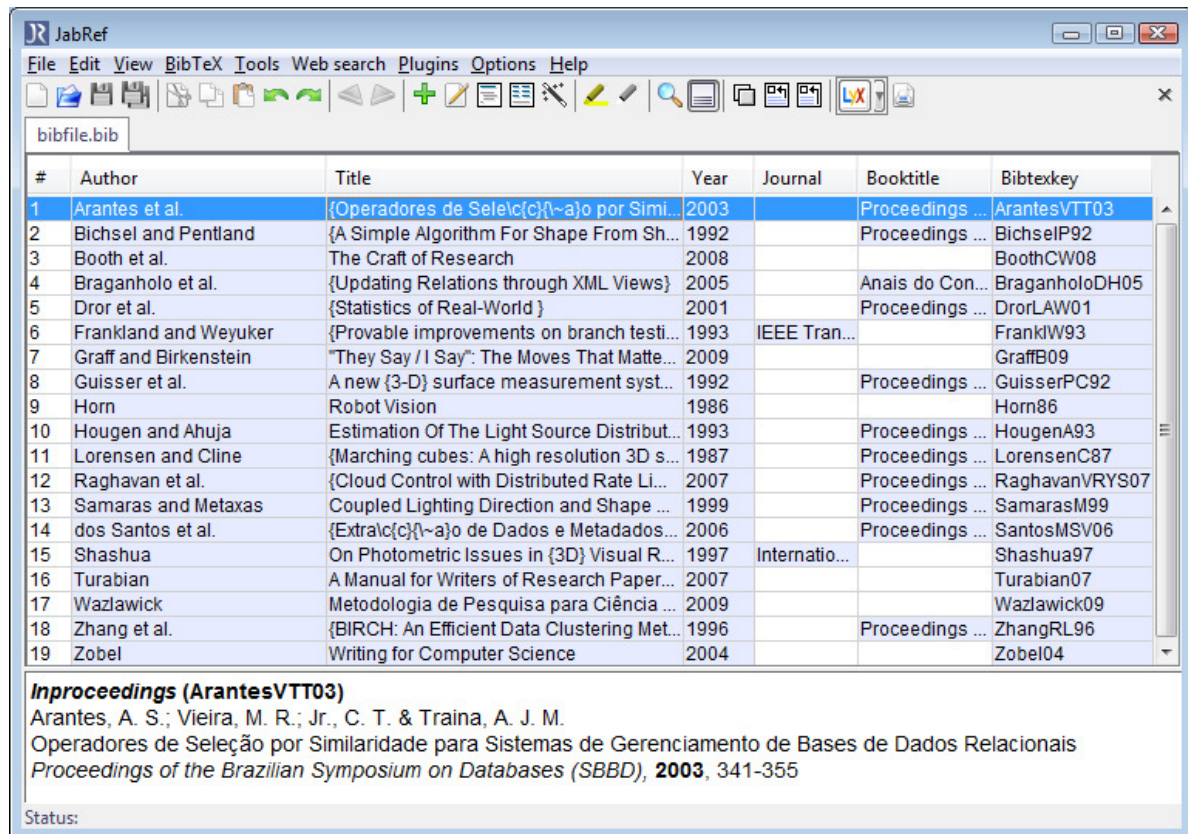


Figura 21: Tela do JabRef para uma versão inicial do arquivo bib deste documento

- Permite exportar para diferentes formatos, inclusive para um banco de dados utilizando SQL;
- Tem botão para procurar o artigo da respectiva referência e fazer o seu download;
- Permite adicionar comentários próprios para cada entrada;
- Pode-ser classificar as referências e criar grupos para as mesmas, e muito mais.

Listas Listas podem ser definidas com *bullets* ou com números, conforme os exemplos a seguir.

```
\begin{itemize}
\item Item 1 com bullet
\item Item 2 com bullet
\end{itemize}
```

```
\begin{enumerate}
\item Item 1 numerado
\item Item 2 numerado
\end{enumerate}
```

Fontes Coloridas Para adicionar texto em cores (muito útil para marcar trechos do texto que estão *em trabalho*, deve-se adicionar os pacotes *graphicx* e *color* (usando o comando `\usepackage` e depois utilizar o comando `\textcolor{cor}{texto}` para colorir o *texto* com a *cor* especificada. Por exemplo `\textcolor{blue}{texto em azul}`. Outras cores comuns são *red* e *green*.

Para Economizar Espaço Existem alguns *dirty tricks*⁴ pra economizar espaço, como por exemplo:

- `\usepackage{times}` Usa fonte *Times* no lugar da default.
- `\usepackage[small,compact]{titlesec}` Modifica o título e os espaços antes/depois dos mesmos.
- `\usepackage[small,it]{caption}` Reduz o tamanho das legendas de tabelas e figuras.

WEB A Web é repleta de páginas e documentos sobre LaTeX. Alguns exemplos incluem:

- Favorito inglês: <<http://en.wikibooks.org/wiki/LaTeX/>>
- Favorito português: <<http://linorg.usp.br/CTAN/info/lshort/portuguese/pt-lshort.pdf>>
- <<http://www.mat.ufmg.br/~regi/topicos/intlat.pdf>>
- <<http://www.duke.edu/~hg9/ctex/LaTeXManual.pdf>>
- <<http://minerva.ufpel.tche.br/~campani/cursolatex.pdf>>
- <<http://www.personal.ceu.hu/tex/words.htm>>

⁴ Ou seja, eles irão alterar a formatação dada pelo estilo default do texto.

Índice

Adobe Illustrator, [50](#)

Adobe Photoshop, [51](#)

alíneas, [54](#)

citações

 diretas, [47](#)

 simples, [48](#)

CorelDraw, [50](#)

espaçamento

 do primeiro parágrafo, [55](#)

 dos parágrafos, [55](#)

 entre as linhas, [56](#)

 entre os parágrafos, [55](#)

expressões matemáticas, [53](#)

Fig., [47](#)

figuras, [49](#)

filosofia, [49](#)

Gimp, [51](#)

incisos, [54](#)

InkScape, [50](#)

PLN, [27](#)

POS, [27](#), [28](#), [40](#)

subalíneas, [54](#)

tabelas, [48](#)

TCC, [47](#)