

# Natural Language Processing

## Dealing with unstructured data

Modelos de Linguagem

Renata Vieira  
Joaquim Neto

Rio de Janeiro - April, 2019

**Material em:**

<https://bit.ly/2YGmYje>

# Cronograma

Tema: Modelos de Linguagem

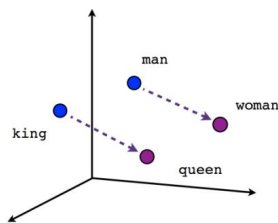
I Dia	II Dia
Contextualização WE	Modelos BERT
Arquiteturas de WE	Flair Embedding
Geração de WE	Geração de Modelos Flair

# Motivação

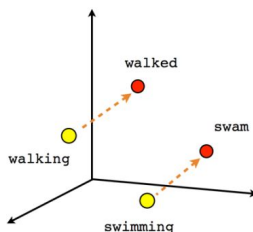
- Seria possível representar uma linguagem natural por um modelo?
- De que natureza é esse modelo?

# Motivação

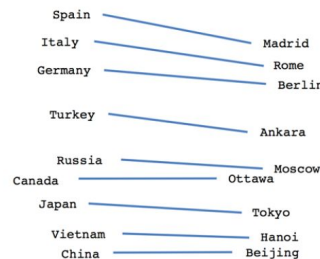
- Seria possível representar uma linguagem natural por um modelo?
- De que natureza é esse modelo?
- SIM, podemos representar uma linguagem natural por um modelo de natureza matemática!



Male-Female



Verb tense



Country-Capital

# Modelos de Linguagem

Algumas aplicações de Word Embedding:

- POS;
- NER (PLN-PUCRS);
- Similaridade Semântica;
- Ontologias (PLN-PUCRS);
- Saúde (PLN-PUCRS);

# Modelos de Linguagem



**Uma limitação:**

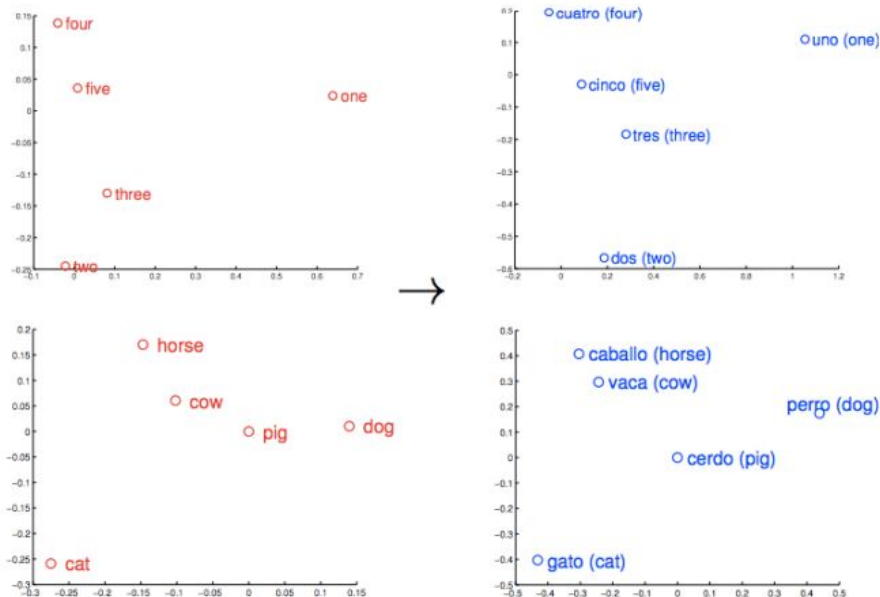
Embeddings Polissêmicos



# Modelos de Linguagem

## Uma característica:

Idiomas diferentes tendem a produzir vetores em locais semelhantes.



# Modelos de Linguagem

De forma mais precisa, um **espaço vetorial sobre  $\mathbb{R}$**  é um conjunto  $V$ , cujos elementos são chamados vetores, equipado com duas operações:

## Uma característica:

Podemos fazer operações com palavras!

- Similaridade (Cos);
- Soma e Produto;
- Entre outras;

- Soma entre vetores, que satisfaz
  1. Associatividade:  $(u + v) + w = u + (v + w)$ , para quaisquer  $u, v, w \in V$ ;
  2. Elemento neutro: existe o vetor  $0 \in V$  que satisfaz  $v + 0 = 0 + v = v$ , para qualquer  $v \in V$ ;
  3. Inverso aditivo: para cada  $v \in V$ , existe o inverso  $u = -v \in V$ , que satisfaz  $v + u = 0$ ;
  4. Comutatividade:  $u + v = v + u$ , para quaisquer  $u, v \in V$ ;
- Multiplicação de vetor por escalar, que satisfaz
  5. Associatividade da multiplicação por escalar:  $a \cdot (b \cdot v) = (a \cdot b) \cdot v$ , para quaisquer  $a, b \in \mathbb{R}$  e qualquer  $v \in V$ ;
  6. Vale que  $1 \cdot v = v$ , ou seja, a unidade dos números reais não altera os vetores de  $V$ ;
  7. Distributiva de um escalar em relação à soma de vetores:  
 $a \cdot (u + v) = a \cdot v + a \cdot u$ , para qualquer  $a \in \mathbb{R}$  e quaisquer  $u, v \in V$ ;
  8. Distributiva da soma de escalares em relação a um vetor:  
 $(a + b) \cdot v = a \cdot v + b \cdot v$ , para quaisquer  $a, b \in \mathbb{R}$  e qualquer  $v \in V$ .

# Modelos de Linguagem

- Vários trabalhos de PLN têm usado representação vetorial de palavras, normalmente conhecidas como **Neural Embeddings** ou **Word Embeddings** [Levy, 2014];
- Essas representações têm ajudado aos algoritmos resolverem com maior eficiência determinadas tarefas de PLN [Mikolov, 2013];
- Um grupo de algoritmos chamado **Word2vec** tem se destacado em termos de uso e resultados;
- Os algoritmos do **Word2vec** (Redes Neurais), são capazes de aprender as representações vetoriais de palavras contidas em um **espaço vetorial** de alta dimensionalidade [Zhang, 2015];

# Modelos de Linguagem

- Existem várias arquiteturas de Rede Neural que geram Embeddings:

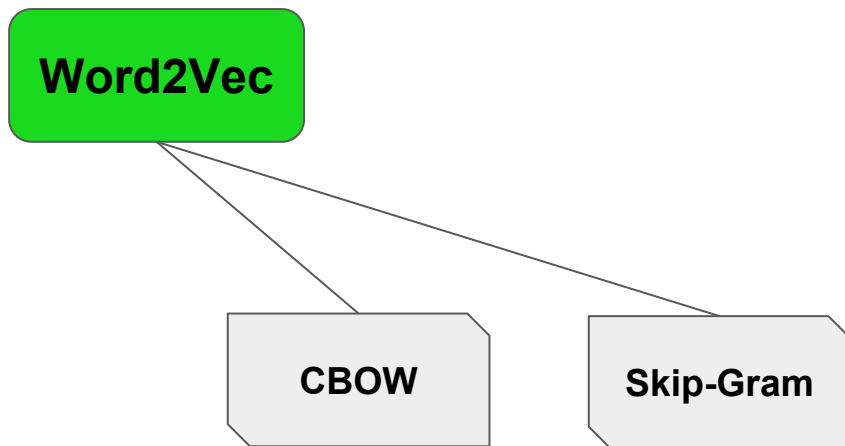
# Modelos de Linguagem

- Existem várias arquiteturas de Rede Neural que geram Embeddings:

**Word2Vec**

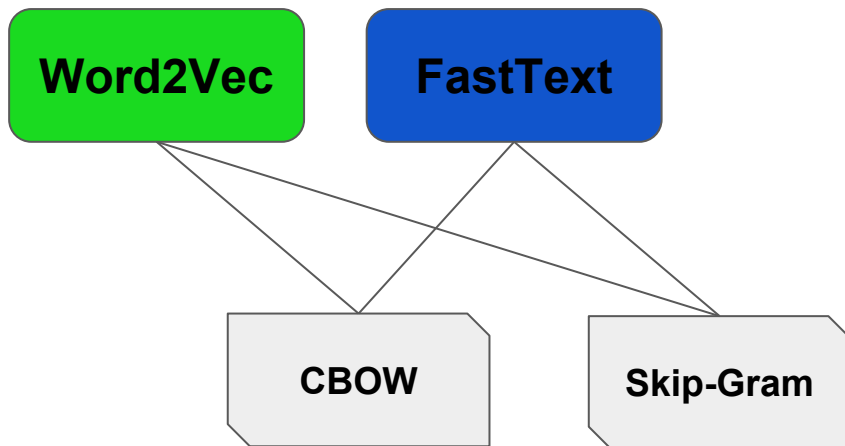
# Modelos de Linguagem

- Existem várias arquiteturas de Rede Neural que geram Embeddings:



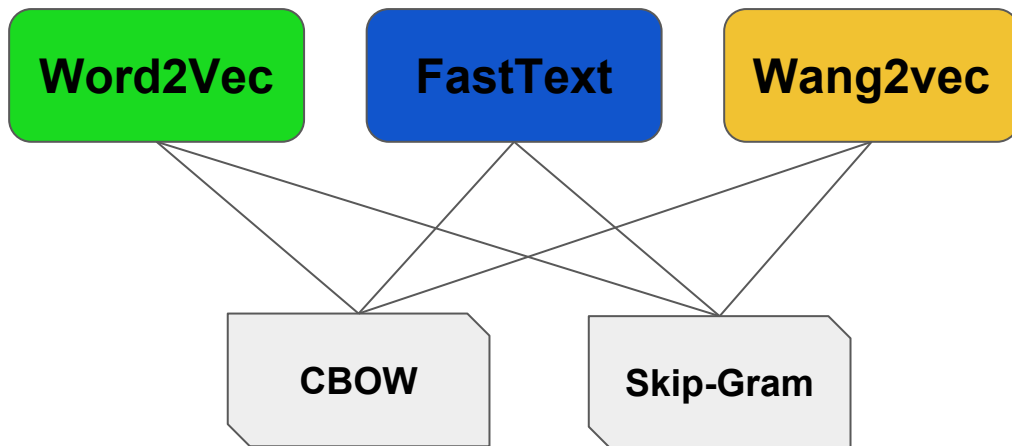
# Modelos de Linguagem

- Existem várias arquiteturas de Rede Neural que geram Embeddings:



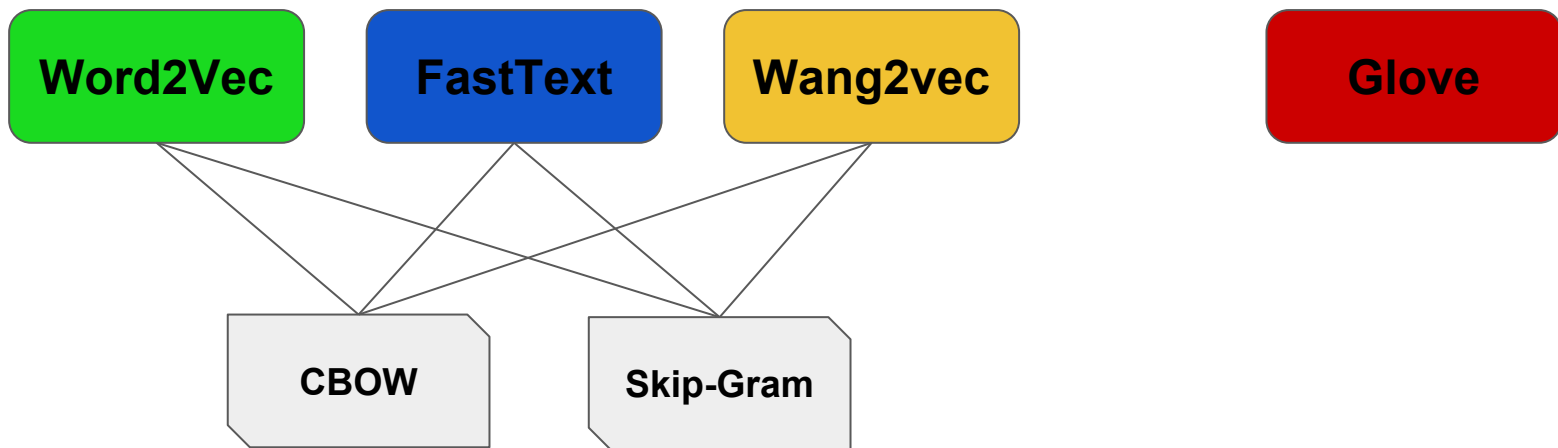
# Modelos de Linguagem

- Existem várias arquiteturas de Rede Neural que geram Embeddings:



# Modelos de Linguagem

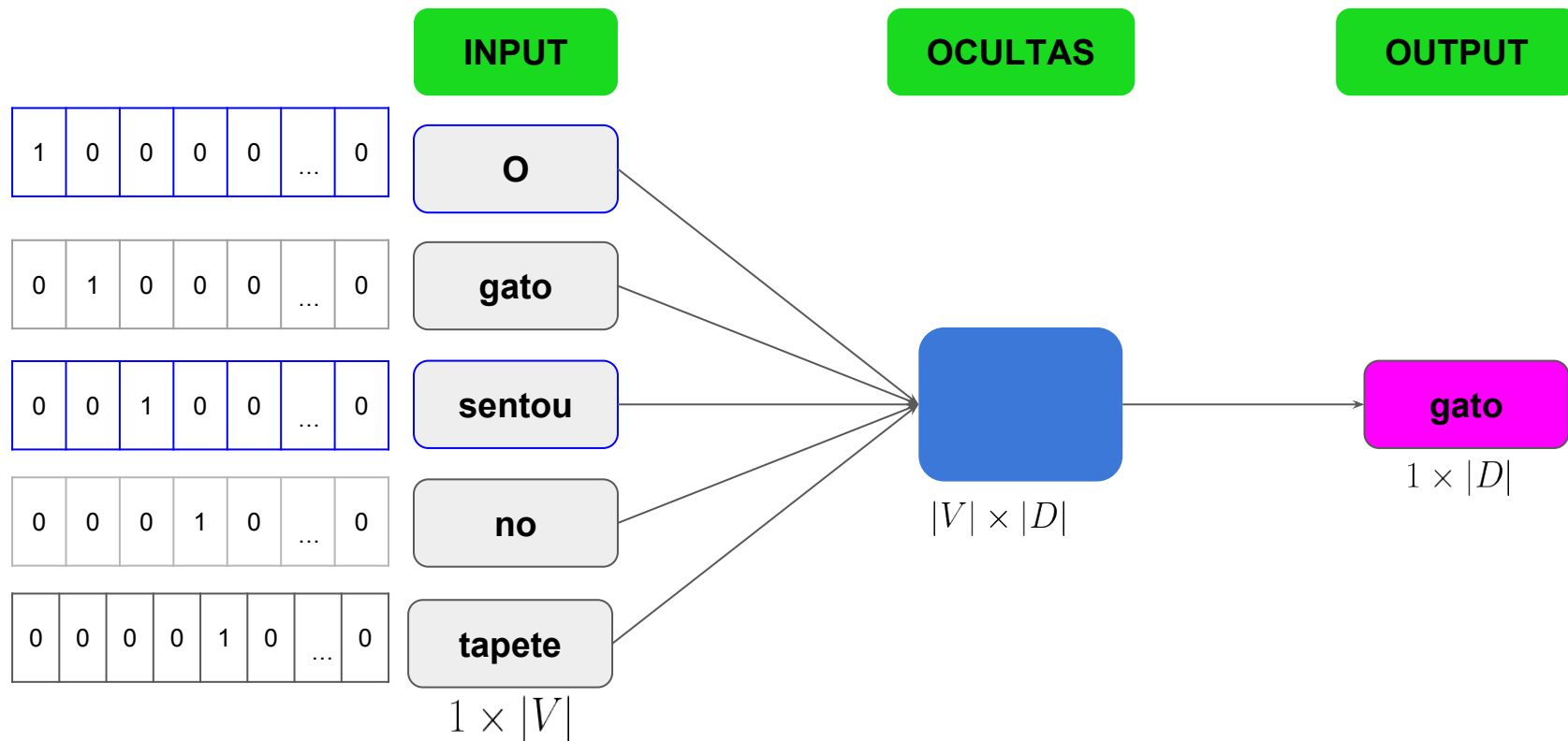
- Existem várias arquiteturas de Rede Neural que geram Embeddings:



# Modelos de Linguagem

CBOW - Continuous Bag-Of-Words

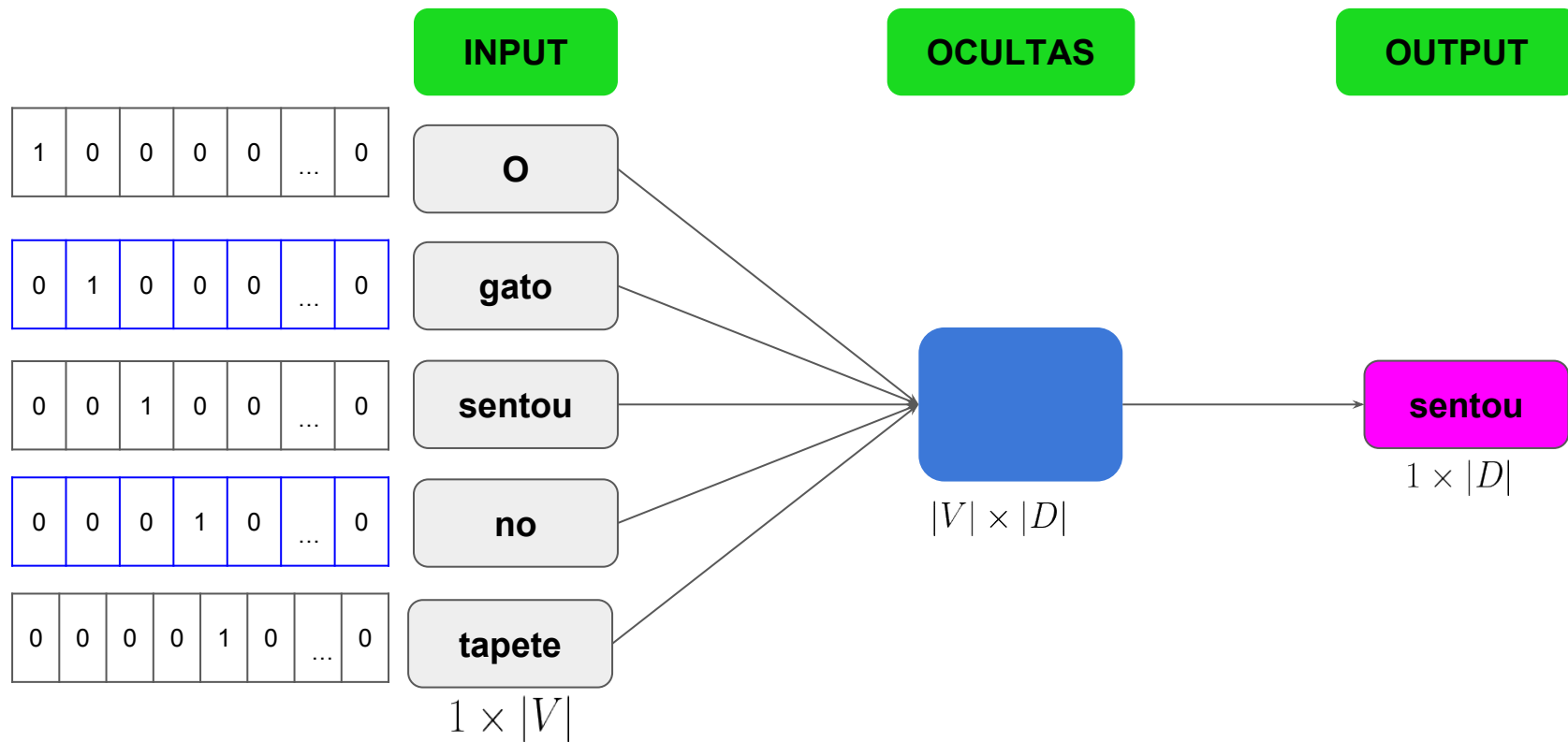
O gato sentou no tapete



# Modelos de Linguagem

CBOW - Continuous Bag-Of-Words

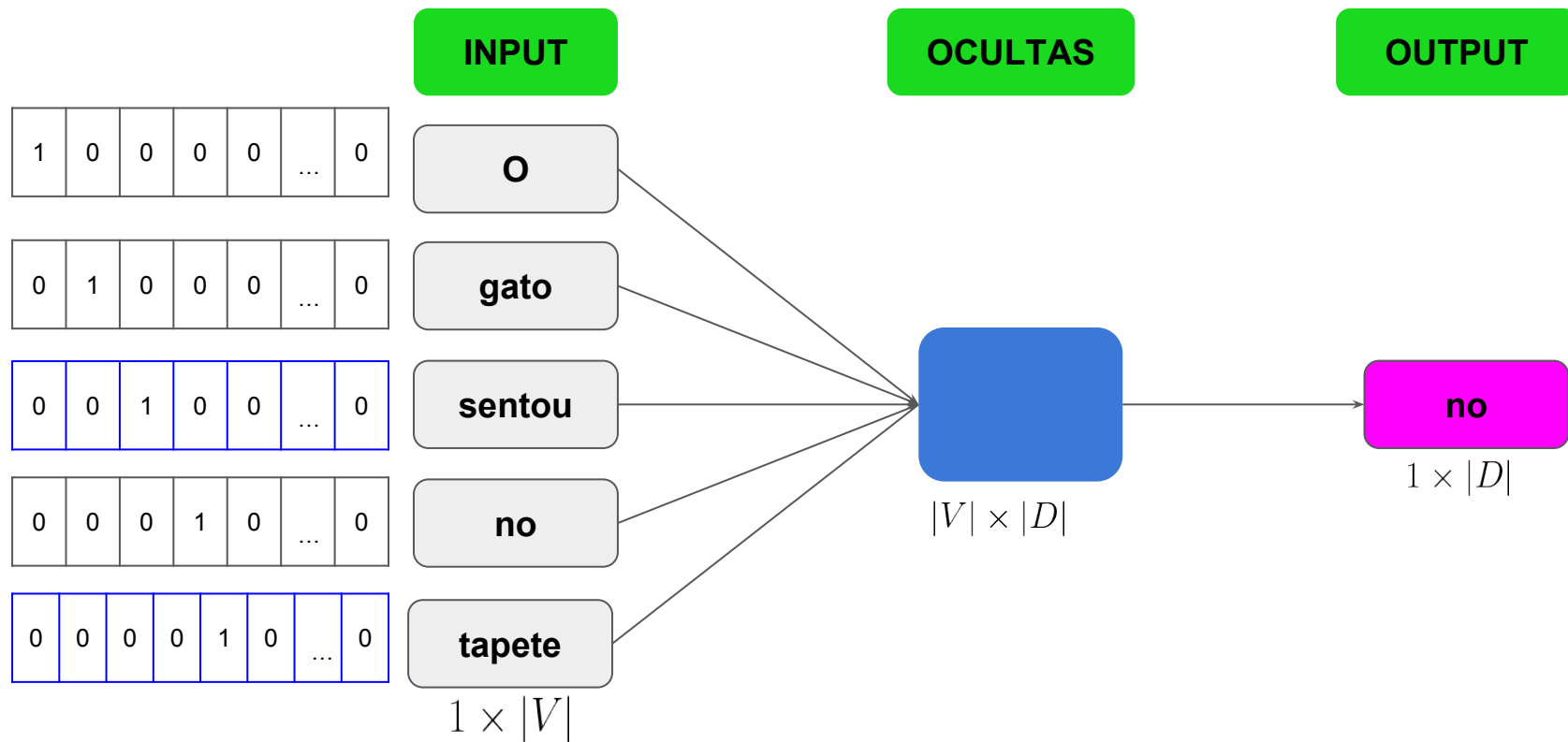
O gato sentou no tapete



# Modelos de Linguagem

CBOW - Continuous Bag-Of-Words

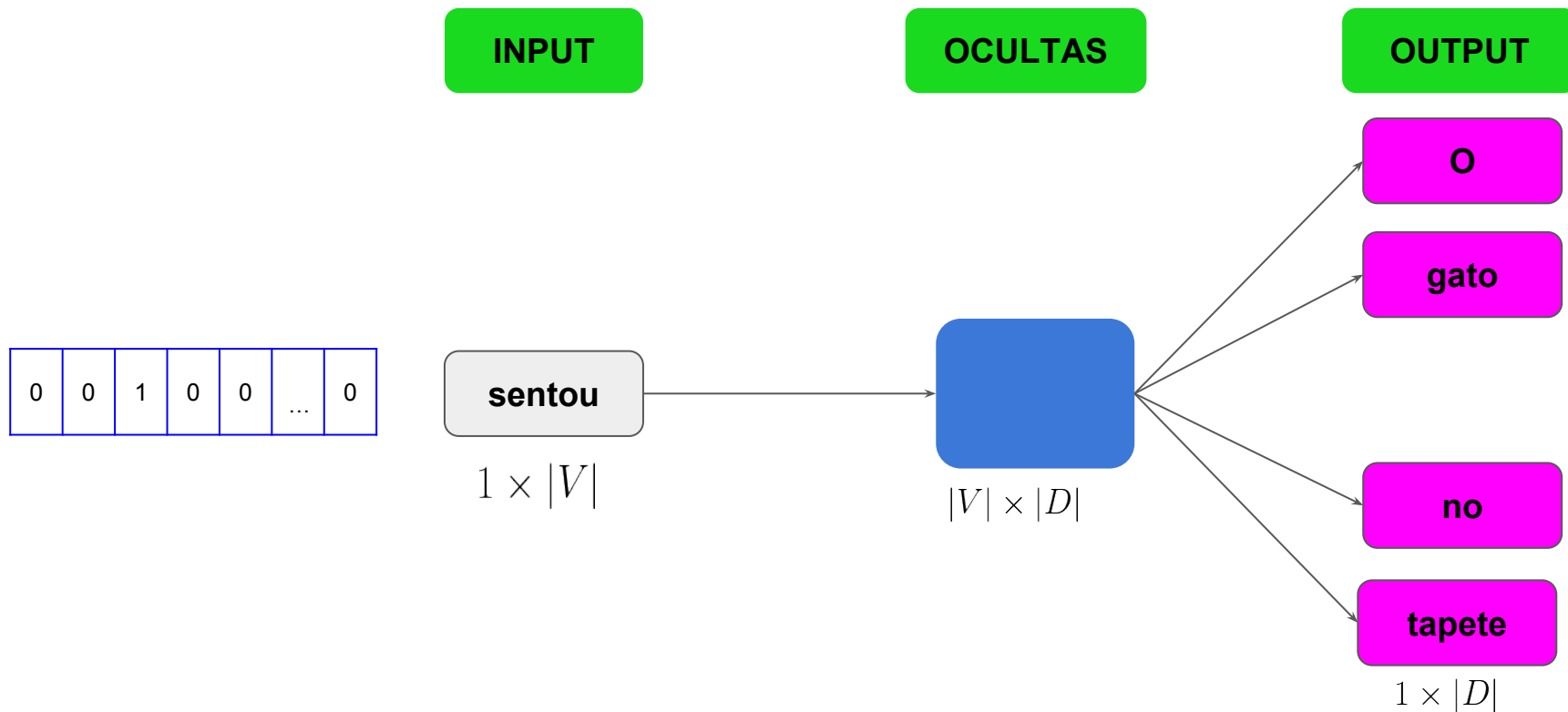
O gato sentou no tapete



# Modelos de Linguagem

Skip-Gram

O gato **sentou** no tapete



# Modelos de Linguagem

## Word2Vec

- Desenvolvido por pesquisadores da Google;
- Possibilidade de dois tipos de treinamento:
  - Skip-Gram
  - CBOW

# Modelos de Linguagem

## FastText

- Desenvolvido por pesquisadores do Facebook;
- Consegue aproximar um embedding para palavras não contidas no vocabulário;
- Cada palavra é representado por uma coleção de n-gramas:

```
renoir = <re, ren, eno, noi, oir,  
ir>
```

- FastText tem sido usado para diversas tarefas, entre elas classificação de texto e REN [Bojanowski, 2016];

# Modelos de Linguagem

## Wang2Vec

- Desenvolvido por pesquisadores do INESC-ID (Portugal) e CMU (EUA);
- Modificação no Word2Vec (Skip-Gram e CBOW) para tarefas envolvendo sintaxe;
- Impulsionaram resultados para Part-Of-Speech Tagging e Dependency Parsing [Ling, 2015];

# Modelos de Linguagem



## **Glove**

- Desenvolvido por pesquisadores de Stanford;
- Utiliza uma Matriz de co-ocorrência entre palavras em um contexto para aprender seu significado;
- As probabilidades de predição são calculadas com base na matriz de co-ocorrência [Pennington, 2014];

# Modelos de Linguagem

Corpus	Tokens	Types	Genre	Description
LX-Corpus [Rodrigues et al. 2016]	714,286,638	2,605,393	Mixed genres	A huge collection of texts from 19 sources. Most of them are written in European Portuguese.
Wikipedia	219,293,003	1,758,191	Encyclopedic	Wikipedia dump of 10/20/16
GoogleNews	160,396,456	664,320	Informative	News crawled from GoogleNews service
SubIMDB-PT	129,975,149	500,302	Spoken language	Subtitles crawled from IMDb website
G1	105,341,070	392,635	Informative	News crawled from G1 news portal between 2014 and 2015.
PLN-Br [Bruckshen et al. 2008]	31,196,395	259,762	Informative	Large corpus of the PLN-BR Project with texts sampled from 1994 to 2005. It was also used by [Hartmann 2016] to train word embeddings models
Literacy works of public domain	23,750,521	381,697	Prose	A collection of 138,268 literary works from the Domínio Público website
Lacio-web [Alufio et al. 2003]	8,962,718	196,077	Mixed genres	Texts from various genres, e.g., literary and its subdivisions (prose, poetry and drama), informative, scientific, law, didactic technical
Portuguese e-books	1,299,008	66,706	Prose	Collection of classical fiction books written in Brazilian Portuguese crawled from Literatura Brasileira website
Mundo Estranho	1,047,108	55,000	Informative	Texts crawled from Mundo Estranho magazine
CHC	941,032	36,522	Informative	Texts crawled from Ciência Hoje das Crianças (CHC) website
FAPESP	499,008	31,746	Science Communication	Brazilian science divulgation texts from Pesquisa FAPESP magazine
Textbooks	96,209	11,597	Didactic	Texts for children between 3rd and 7th-grade years of elementary school
Folhinha	73,575	9,207	Informative	News written for children, crawled in 2015 from Folhinha issue of Folha de São Paulo newspaper
NILC subcorpus	32,868	4,064	Informative	Texts written for children of 3rd and 4th-years of elementary school
Para Seu Filho Ler	21,224	3,942	Informative	News written for children, from Zero Hora newspaper
SARESP	13,308	3,293	Didactic	Text questions of Mathematics, Human Sciences, Nature Sciences and essay writing to evaluate students
<b>Total</b>	1,395,926,282	3,827,725		

Corpora STIL 2017

# Modelos de Linguagem

Modelos para o Português disponíveis no repositório do NILC (USP - São Carlos)

NILC - Núcleo Interinstitucional de Linguística Computacional

Acesso: <http://nilc.icmc.usp.br/embeddings>

## Word2Vec

Modelo

CBOW 50 dimensões  
CBOW 100 dimensões  
CBOW 300 dimensões  
CBOW 600 dimensões  
CBOW 1000 dimensões  
SKIP-GRAM 50 dimensões  
SKIP-GRAM 100 dimensões  
SKIP-GRAM 300 dimensões  
SKIP-GRAM 600 dimensões  
SKIP-GRAM 1000 dimensões

[Ver Detalhes »](#)

Corpora STIL 2017

[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)

## Wang2Vec

Modelo

CBOW 50 dimensões  
CBOW 100 dimensões  
CBOW 300 dimensões  
CBOW 600 dimensões  
CBOW 1000 dimensões  
SKIP-GRAM 50 dimensões  
SKIP-GRAM 100 dimensões  
SKIP-GRAM 300 dimensões  
SKIP-GRAM 600 dimensões  
SKIP-GRAM 1000 dimensões

Corpora STIL 2017

[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)

## FastText

Modelo

CBOW 50 dimensões  
CBOW 100 dimensões  
CBOW 300 dimensões  
CBOW 600 dimensões  
CBOW 1000 dimensões  
SKIP-GRAM 50 dimensões  
SKIP-GRAM 100 dimensões  
SKIP-GRAM 300 dimensões  
SKIP-GRAM 600 dimensões  
SKIP-GRAM 1000 dimensões

[Ver Detalhes »](#)

Corpora STIL 2017

[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)

## Glove

Modelo

GLOVE 50 dimensões  
GLOVE 100 dimensões  
GLOVE 300 dimensões  
GLOVE 600 dimensões  
GLOVE 1000 dimensões

[Ver Detalhes »](#)

Corpora STIL 2017

[download](#)  
[download](#)  
[download](#)  
[download](#)  
[download](#)

# Modelos de Linguagem

Como treinar Modelos Word Embedding?

- [Gensim](#) biblioteca para Python: *W2V*, *FT*;
  - Corpora: +1Bi tokens;
  - `sg=1, size=50~1000 (300), window=5, min_count=5~10`
- Glove no [GitHub](https://github.com/stanfordnlp/glove) (<https://github.com/stanfordnlp/glove>);

**Fim do Primeiro Dia!**

# Modelos de Linguagem

BERT - Bidirectional Encoder Representations from Transformers [Devlin, 2018]

- Recente modelo de representação de linguagem desenvolvido pelo Google;
- No artigo original de apresentação do BERT são apresentadas 11 tarefas de PLN onde o BERT foi aplicado; entre elas:
  - **Multi-Genre Natural Language Inference:** dado um par de sentenças deve-se decidir se a segunda sentença é uma continuação, contradição ou neutra em relação a primeira sentença;
  - **Semantic Textual Similarity Benchmark:** classifica a similaridade entre duas sentenças;
  - **Named Entity Recognition:** reconhecer em um texto determinadas menções e classificá-las em categorias;

# Modelos de Linguagem

- Treinar um modelo BERT envolve duas tarefas:

**Masked LM**

**Next Sentence Prediction**

# Modelos de Linguagem

**Masked LM:** o objetivo é "mascarar" 15% dos tokens de uma sentença para predição. Algumas regras:

- 80% das vezes: a palavra é substituída pelo símbolo [MASK], por exemplo:

meu cachorro é cabeludo. → meu [MASK] é cabeludo.

- 10% das vezes: a palavra é substituída por uma palavra aleatória, por exemplo:

meu cachorro é cabeludo. → meu cachorro é maçã.

- 10% das vezes: a palavra é mantida, por exemplo:

meu cachorro é cabeludo. → meu cachorro é cabeludo.

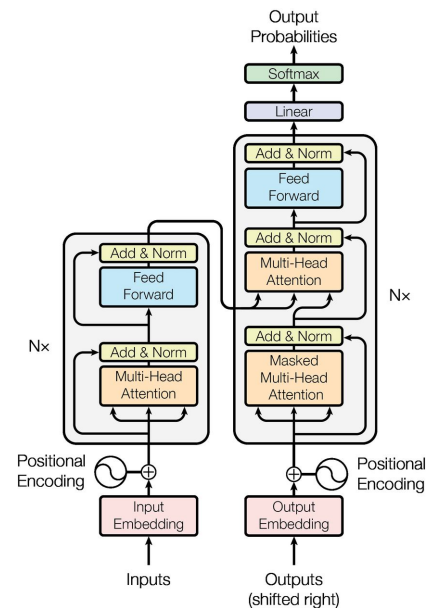
# Modelos de Linguagem

**Next Sentence Prediction:** dado um par de sentenças  $(s_1, s_2)$  o modelo deve prever se a sentença  $s_2$  é a subsequente a  $s_1$ . Algumas regras:

- 50% dos pares de sentenças são tal que  $s_2$  é de fato a sentença subsequente;
- 50% dos outros pares são tal que  $s_2$  é uma sentença aleatória do corpus;

# Modelos de Linguagem

A Rede Neural responsável por realizar as tarefas **Masked LM** e **Next Sentence Prediction** é composta por várias camadas do neurônio **Transformer** [Vaswani, 2017].



# Modelos de Linguagem

- Treinar um modelo de linguagem BERT requer muito recurso computacional;
- Por exemplo, para o Inglês foram usadas 16 Cloud TPUs com um corpora de 3,3Bi tokens;
- Existe um modelo Multi-lingue que contempla o **Português**;

TensorFlow:

<https://github.com/google-research/bert>

PyTorch:

<https://github.com/huggingface/pytorch-pretrained-BERT>

Tipo	Idioma	Camadas	Camadas Ocultas	Self-Attention heads
BERT <sub>BASE</sub>	Inglês	12	768	12
BERT <sub>LARGE</sub>	Inglês	24	1024	16
BERT <sub>BASE</sub>	Multi-lingue	12	768	12

# Modelos de Linguagem

**Flair Embedding** - é um recente modelo de embedding que permite a modelagem da linguagem com base na distribuição das sequências de caracteres e palavras [Akbik, 2018];

- A geração de um Flair Embedding **não só depende** do contexto das **palavras** vizinhas, mas também do nível de **caractere** das palavras vizinhas;
- O modelo é gerado a partir de duas redes **LSTM** que capturam e incorporam as informações aprendidas durante o treinamento;

# Modelos de Linguagem

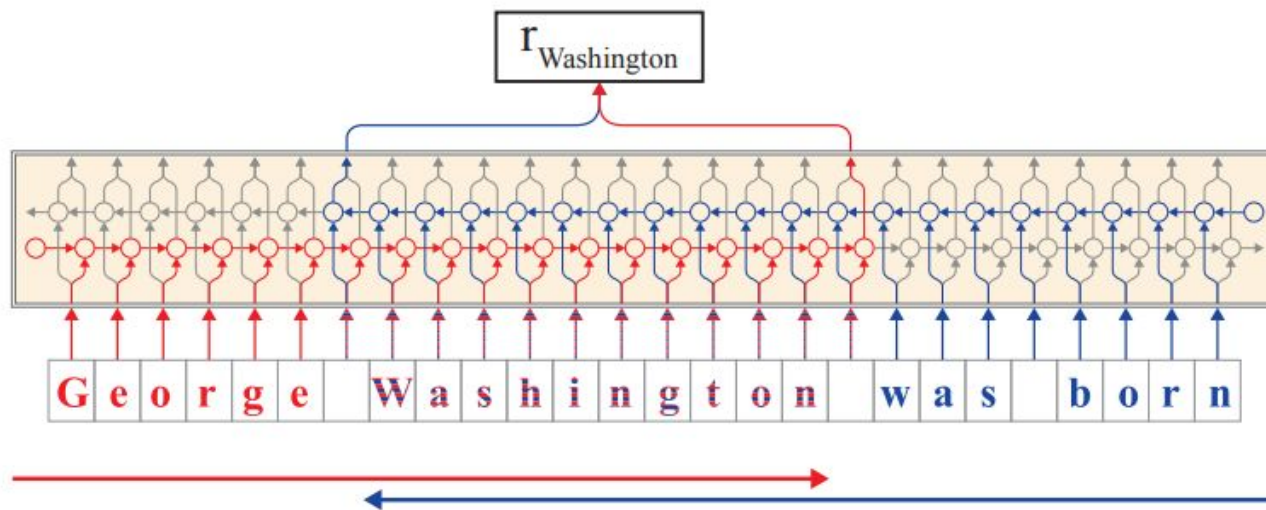
**LSTM - Long Short-Term Memory** - é um tipo de Rede Neural Recorrente, com uma estrutura computacional mais complexa, que tem tido sucesso na resolução de tarefas sequenciais [Tai, 2015];

- Uma LSTM permite manter, alterar ou descartar informações anteriores para relacionar com uma informação atual;

Há uma variação das redes LSTM, que são as **Bidirectional LSTM (Bi-LSTM)**;

- As redes Bi-LSTM consistem de duas LSTM que funcionam em paralelo;

# Modelos de Linguagem



# Modelos de Linguagem

- Dado  $X_{0:T}$  uma sequência de caracteres, que produz uma linguagem natural;

$$X_{0:T} := (x_0, x_1, \dots, x_{t-1}, x_t)$$

- Poderíamos prever  $\{x_t\} \in X_{0:T} | X_{0:t-1} \subset X_{0:T}$  ?
- Sim, podemos aprender como os caracteres dessa linguagem estão distribuídos, abstraindo um modelo de linguagem [Rosenfeld, 2000];
- Então quando dado  $X_{0:t-1}$  queremos prever  $\{x_t\} \in X_{0:T}$  :

$$P(x_t | x_0, x_1, \dots, x_{t-1}) = P(x_{0:T})$$

# Modelos de Linguagem

- Dado  $X_{0:T}$  uma sequência de caracteres, que produz uma linguagem natural;

$$X_{0:T} := (x_0, x_1, \dots, x_{t-1}, x_t)$$

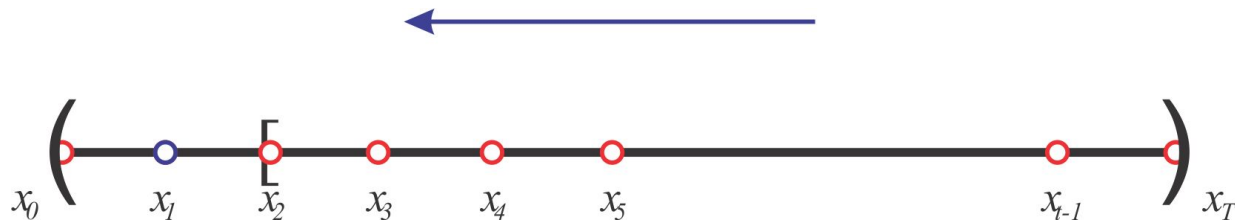
- Poderíamos prever  $\{x_t\} \in X_{0:T} | X_{0:t-1} \subset X_{0:T}$  ?
- **Sim**, podemos aprender como os caracteres dessa linguagem estão distribuídos, abstraindo um modelo de linguagem [Rosenfeld, 2000];
- Então quando dado  $X_{0:t-1}$  queremos prever  $\{x_t\} \in X_{0:T}$  :

$$P(x_t | x_0, x_1, \dots, x_{t-1}) = P(x_{0:T}) \approx \prod_{t=0}^T P(x_t | h_t; \theta)$$

# Modelos de Linguagem

- Então quando dado  $X_{0:t-1}$  queremos prever  $\{x_t\} \in X_{0:T}$  :

$$P(x_t|x_0, x_1, \dots, x_{t-1}) = P(x_{0:T}) \approx \prod_{t=0}^T P(x_t|h_t; \theta)$$



# Modelos de Linguagem

- Como Flair Embedding é treinado por duas LSTM temos dois modelos:

$$p^f(x_t|X_{t+1:T}) \approx \prod_{t=0}^T p^f(x_t|h_t^f; \theta)$$

$$h_t^f = f_h^f(x_{t-1}, h_{t-1}^f, c_{t-1}^f; \theta)$$

$$c_t^f = f_c^f(x_{t-1}, h_{t-1}^f, c_{t-1}^f; \theta)$$

**Forward**

$$p^b(x_t|X_{t+1:T}) \approx \prod_{t=0}^T p^b(x_t|h_t^b; \theta)$$

$$h_t^b = f_h^b(x_{t-1}, h_{t-1}^b, c_{t-1}^b; \theta)$$

$$c_t^b = f_c^b(x_{t-1}, h_{t-1}^b, c_{t-1}^b; \theta)$$

**Backward**

# Modelos de Linguagem

- Os processos de **Forward** e **Backward** produzem duas saídas para cada palavra:  $h_t^f$  e  $h_t^b$  ;
- Essas saídas são empilhadas em uma matriz, formando o embedding final:

$$w^{CharLM} := \begin{bmatrix} h_{t_{i+1}-1}^f \\ h_{t_i-1}^b \end{bmatrix}$$

# Modelos de Linguagem

- Existem modelos Flair Embedding para o português disponível para uso:

<https://github.com/zalandoresearch/flair>

- Como podemos treinar um modelo Flair?
- Como podemos aplicar o modelo?

# REN - Reconhecimento de Entidades Nomeadas

- ▶ Reconhecimento de Entidades Nomeadas (REN), é a tarefa de encontrar nomes próprios em um dado texto e classificá-los entre várias categorias de interesse ou uma categoria padrão chamada Outros [15];
- ▶ As categorias mais comuns são: Pessoa, Organização, Local e Outros;

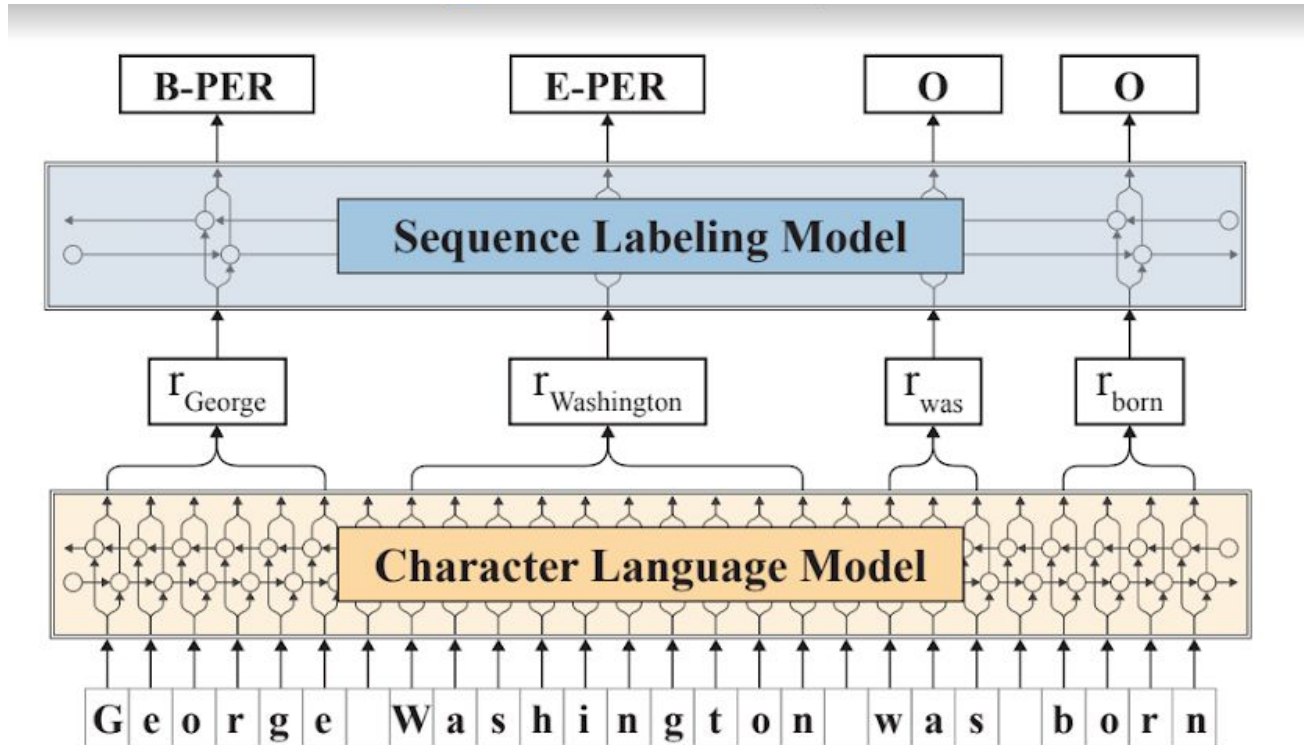
Ex.: Vincent Willem van Gogh nasceu no dia 30 de março de 1853 em Zundert na província predominantemente católica de Brabante do Norte no sul dos Países Baixos. Era o filho mais velho sobrevivente de Anna Cornelia Carbentus e Theodorus van Gogh, um pastor da Igreja Reformada Neerlandesa. <sup>1</sup>

Pessoa — Local — Organização — Outros

---

<sup>1</sup><https://cloud.google.com/natural-language/>

# Flair Embedding para REN



# Referências

**[Akbik, 2018]** Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 1638-1649).

**[Bojanowski, 2016]** Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. arXiv preprint arXiv:1607.04606.

**[Devlin, 2018]** Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

**[Levy, 2014]** Levy, O.; Goldberg, Y. “Dependency-based word embeddings”. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 302–308.

# Referências

**[Ling, 2015]** Ling, W., Dyer, C., Black, A., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.

**[Mikolov, 2013]** Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J. “Distributed representations of words and phrases and their compositionality”. In: Advances in neural information processing systems, 2013, pp. 3111–3119.

**[Pennington, 2014]** Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

**[Rosenfeld, 200]** Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here?. Proceedings of the IEEE, 88(8), 1270-1278.

**[Zhang, 2015]** Zhang, D.; Xu, H.; Su, Z.; Xu, Y. “Chinese comments sentiment classification based on word2vec and svm perf”, Expert Systems with Applications, vol. 42–4, 2015, pp.1857–1863.