PBG0018 Processamento de Linguagem Natural

Prof. Thiago A. S. Pardo

ICMC USP SÃO CARLOS
50 anos 1971 - 2021
Instituto de Ciências Matemáticas e de Computação
| Universidade de São Paulo |

Relembrando

- Córpus: história e definições
- Dados e modelagem matemática e estatística da língua
  - Zipf e abordagens frequentistas
    - Praticidade e limitações
  - Probabilidades
    - Tradução e modelagem de língua
  - Hipótese distribucional e *word embeddings*
    - Vetores esparsos vs densos
    - As bases da revolução recente em PLN

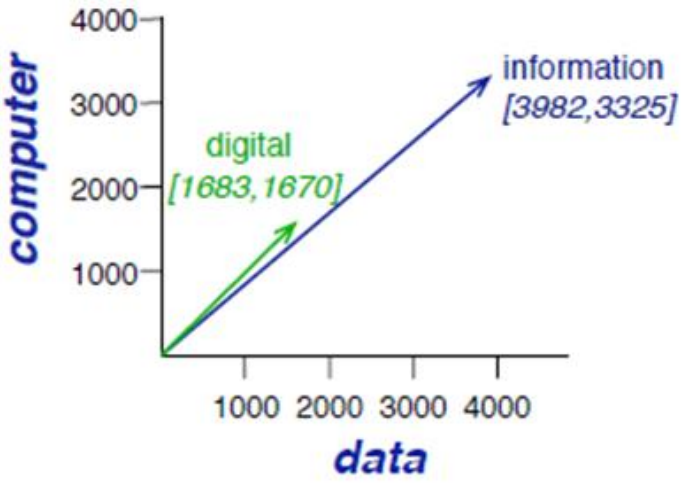
Relembrando

- Matriz termo-contexto

	...	computer	data	...	...
cherry	...	2	8		...
strawberry	...	0	0		...
digital	...	1670	1683	...	...
information	...	3325	3982		...

A palavra "digital" pode ser descrita pelo vetor [..., 1670, 1683, ...]

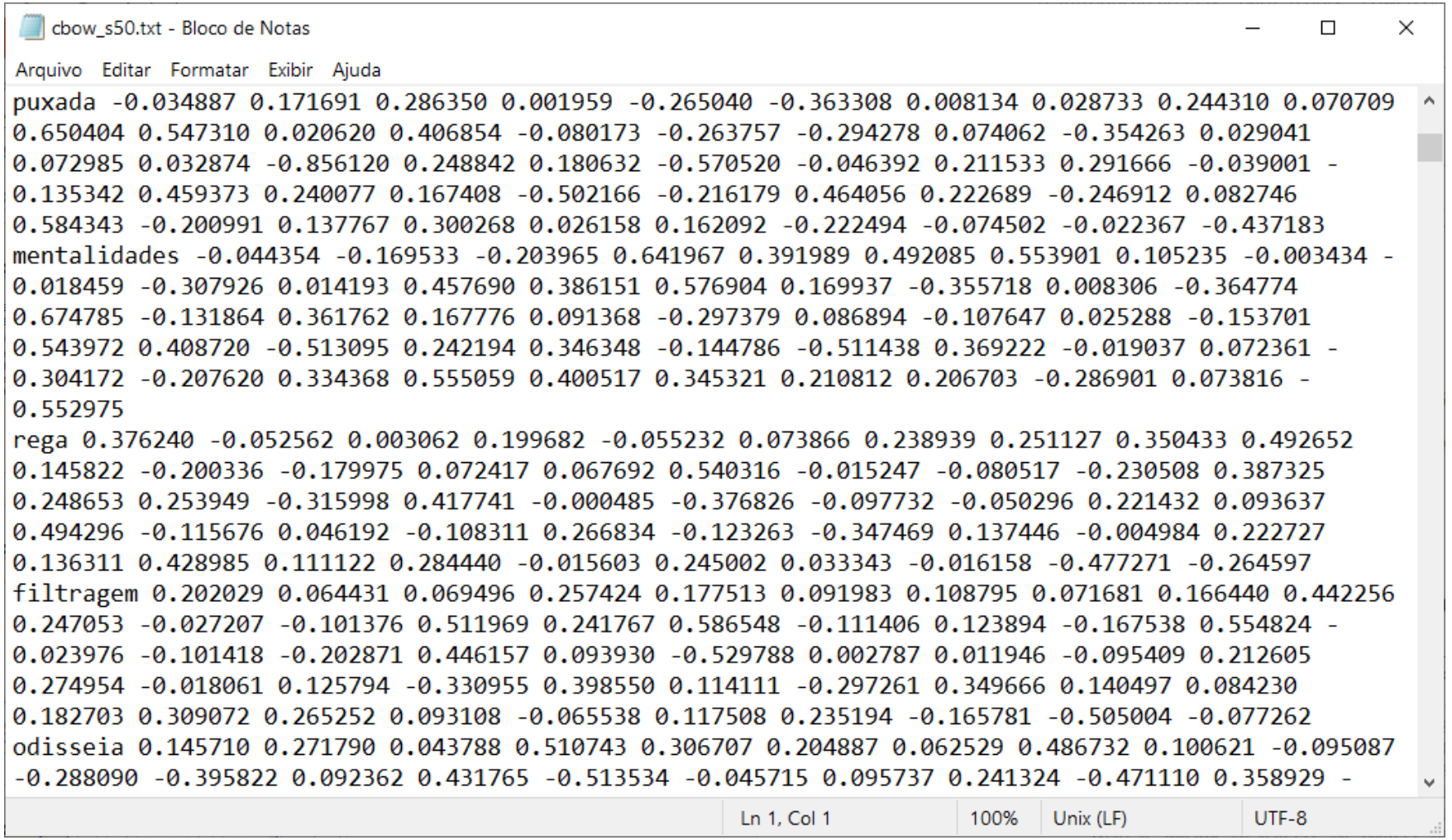
Considerando apenas 2 dimensões: *data* (eixo x) e *computer* (eixo y)



O pioneiro da mudança recente em PLN

- Word2Vec (Mikolov et al., 2013): grande impacto na área
  - "Numerificação" de textos
    - E a compatibilidade da ideia com a proposta das redes neurais artificiais
  - Possibilidade de fazer contas com símbolos
    - "rei" – "homem" + "mulher" = "rainha"
  - Demonstração da força das empresas em PLN

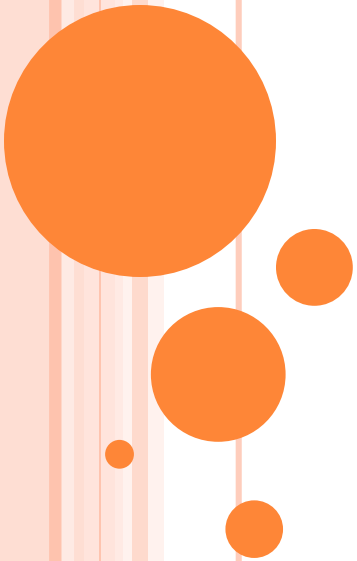
O PIONEIRO DA MUDANÇA RECENTE EM PLN


cbow_s50.txt - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
puxada -0.034887 0.171691 0.286350 0.001959 -0.265040 -0.363308 0.008134 0.028733 0.244310 0.070709
0.650404 0.547310 0.020620 0.406854 -0.080173 -0.263757 -0.294278 0.074062 -0.354263 0.029041
0.072985 0.032874 -0.856120 0.248842 0.180632 -0.570520 -0.046392 0.211533 0.291666 -0.039001 -
0.135342 0.459373 0.240077 0.167408 -0.502166 -0.216179 0.464056 0.222689 -0.246912 0.082746
0.584343 -0.200991 0.137767 0.300268 0.026158 0.162092 -0.222494 -0.074502 -0.022367 -0.437183
mentalidades -0.044354 -0.169533 -0.203965 0.641967 0.391989 0.492085 0.553901 0.105235 -0.003434 -
0.018459 -0.307926 0.014193 0.457690 0.386151 0.576904 0.169937 -0.355718 0.008306 -0.364774
0.674785 -0.131864 0.361762 0.167776 0.091368 -0.297379 0.086894 -0.107647 0.025288 -0.153701
0.543972 0.408720 -0.513095 0.242194 0.346348 -0.144786 -0.511438 0.369222 -0.019037 0.072361 -
0.304172 -0.207620 0.334368 0.555059 0.400517 0.345321 0.210812 0.206703 -0.286901 0.073816 -
0.552975
rega 0.376240 -0.052562 0.003062 0.199682 -0.055232 0.073866 0.238939 0.251127 0.350433 0.492652
0.145822 -0.200336 -0.179975 0.072417 0.067692 0.540316 -0.015247 -0.080517 -0.230508 0.387325
0.248653 0.253949 -0.315998 0.417741 -0.000485 -0.376826 -0.097732 -0.050296 0.221432 0.093637
0.494296 -0.115676 0.046192 -0.108311 0.266834 -0.123263 -0.347469 0.137446 -0.004984 0.222727
0.136311 0.428985 0.111122 0.284440 -0.015603 0.245002 0.033343 -0.016158 -0.477271 -0.264597
filtragem 0.202029 0.064431 0.069496 0.257424 0.177513 0.091983 0.108795 0.071681 0.166440 0.442256
0.247053 -0.027207 -0.101376 0.511969 0.241767 0.586548 -0.111406 0.123894 -0.167538 0.554824 -
0.023976 -0.101418 -0.202871 0.446157 0.093930 -0.529788 0.002787 0.011946 -0.095409 0.212605
0.274954 -0.018061 0.125794 -0.330955 0.398550 0.114111 -0.297261 0.349666 0.140497 0.084230
0.182703 0.309072 0.265252 0.093108 -0.065538 0.117508 0.235194 -0.165781 -0.505004 -0.077262
odisseia 0.145710 0.271790 0.043788 0.510743 0.306707 0.204887 0.062529 0.486732 0.100621 -0.095087
-0.288090 -0.395822 0.092362 0.431765 -0.513534 -0.045715 0.095737 0.241324 -0.471110 0.358929 -
```

Ln 1, Col 1 | 100% | Unix (LF) | UTF-8



Word2Vec

WORD2VEC

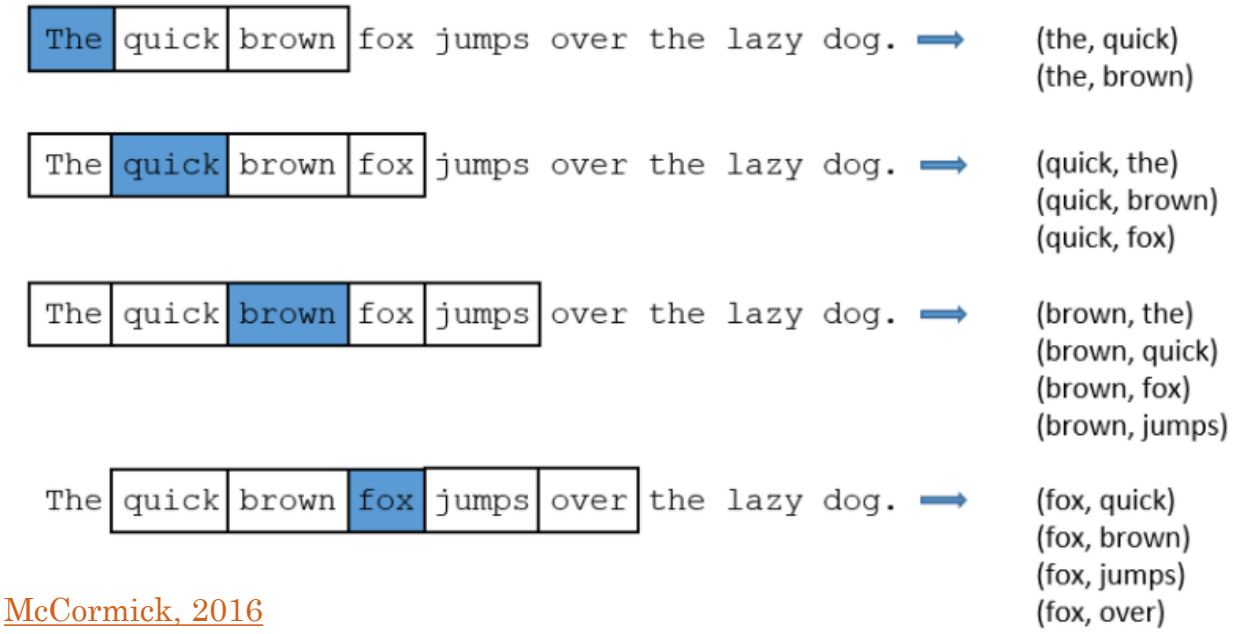
(MIKOLOV ET AL., 2013)

- Revolucionou a área ao apresentar um método relativamente "simples" para aprendizado de vetores densos
  - Com estratégias para otimizar o processo
- Os vetores provêm de "pesos" aprendidos por uma rede neural treinada para uma tarefa "fake"
  - Previsão de uma palavra dado seu contexto
- Na época, o autor principal trabalhava no Google
  - A importância das empresas no avanço recente do PLN!

Dados para aprendizado

- Previsão de palavras que ocorrem no mesmo contexto
  - Intuição: vetores de palavras que ocorrem com o mesmo contexto ("janela") tendem a convergir para valores próximos durante o aprendizado
  - Dados de treino facilmente acessíveis e abundantes!

Exemplo: geração de dados de treino considerando uma janela de +- 2 palavras

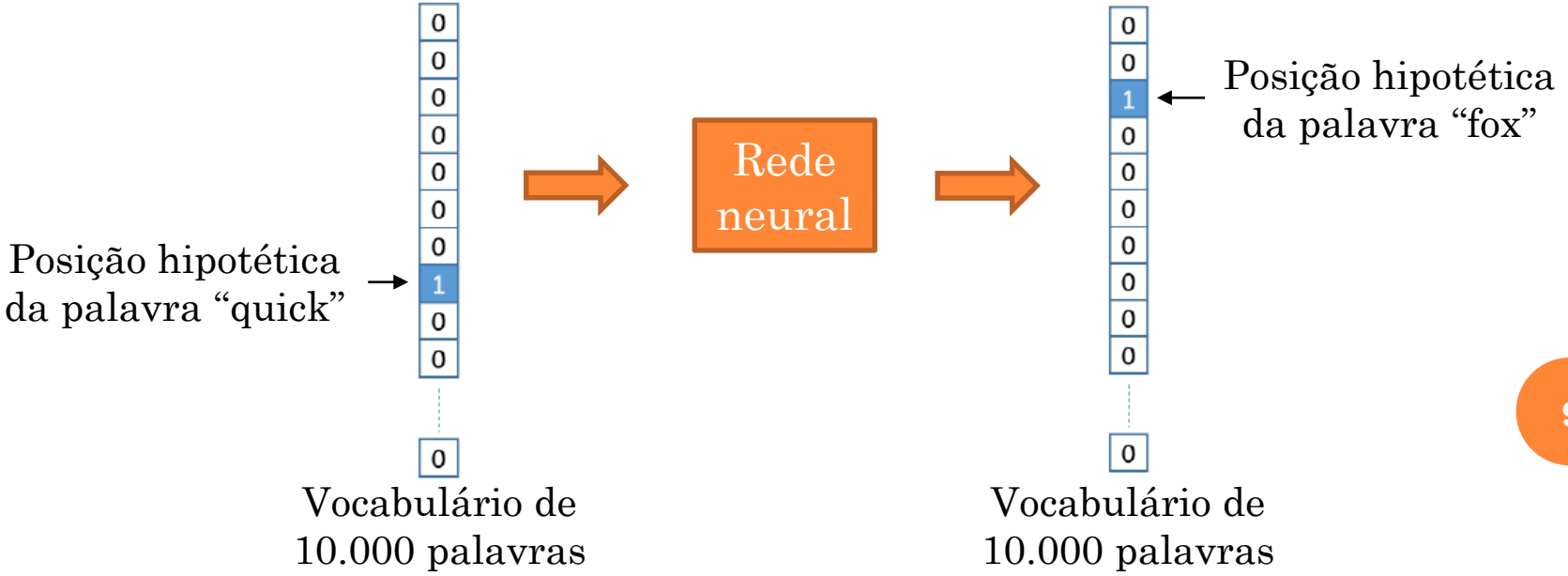


McCormick, 2016

Representação de entrada e saída

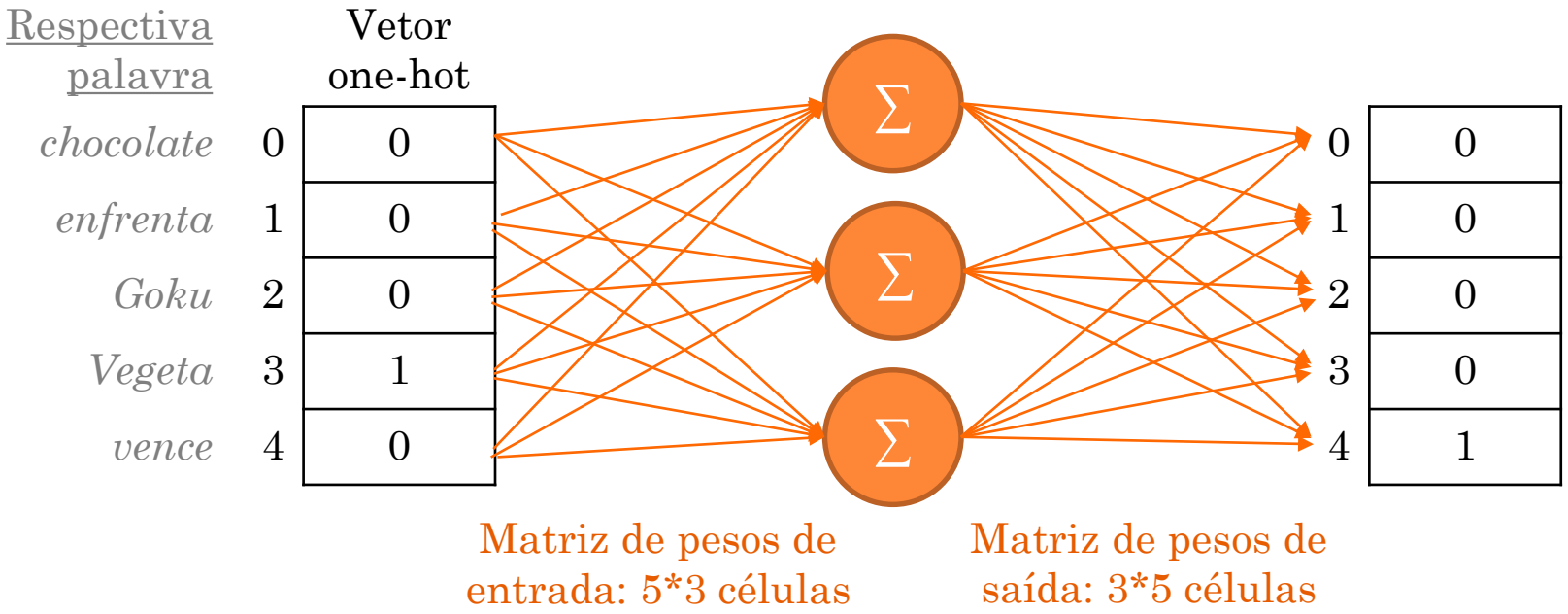
- Vetores one-hot para vocabulário
  - Vetor de cada palavra tem o tamanho do vocabulário
  - A palavra de interesse é marcada com ‘1’, enquanto as demais com ‘0’

- Idealmente, supondo o aprendizado do par (quick, fox) em um vocabulário de 10.000 palavras

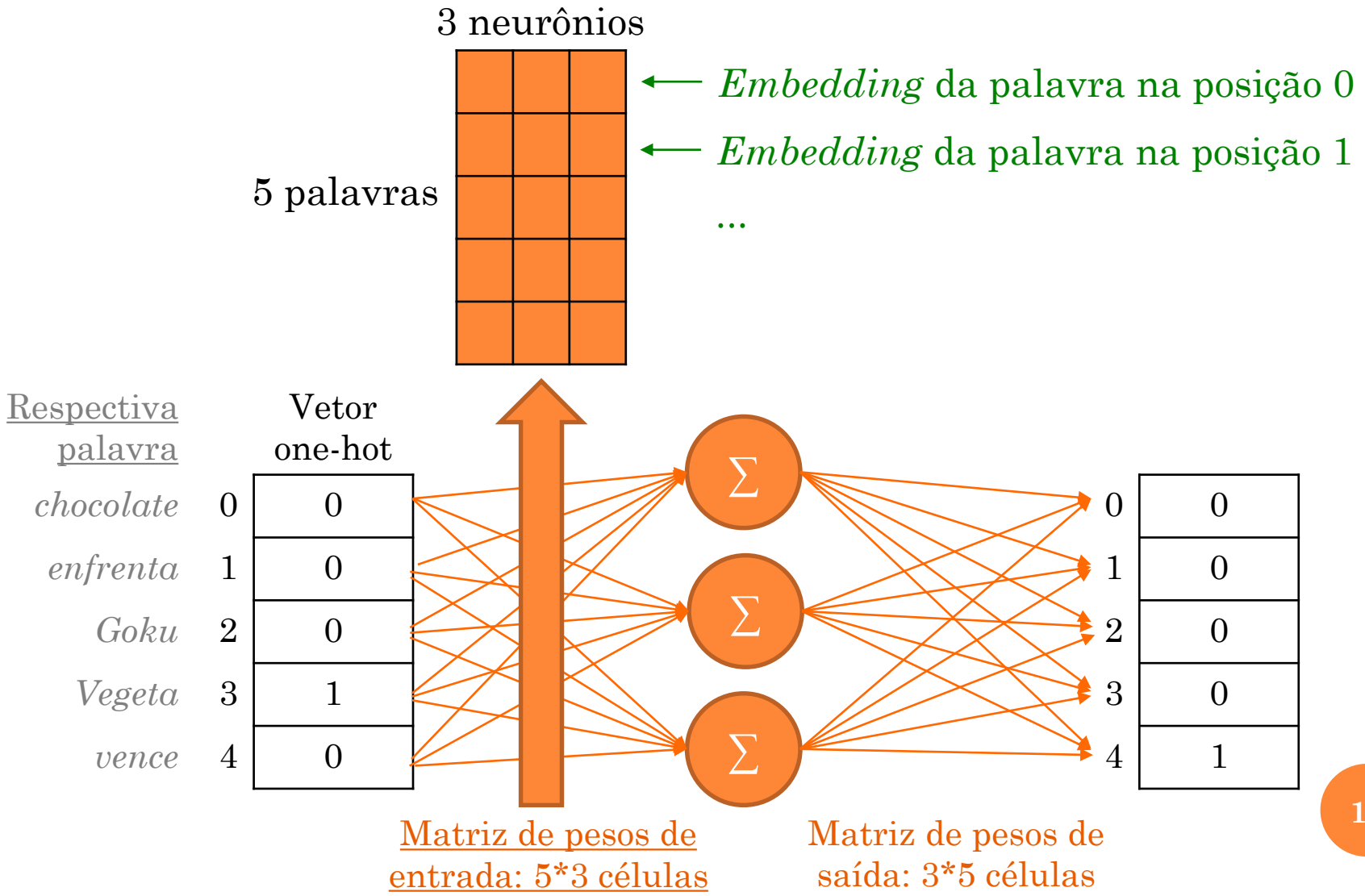


"Abrindo" a rede neural

- Exemplo hipotético simples (totalmente irreal, apenas para fins didáticos)
  - Língua com vocabulário de 5 palavras
  - 3 neurônios na camada escondida da rede



"Abrindo" a rede neural

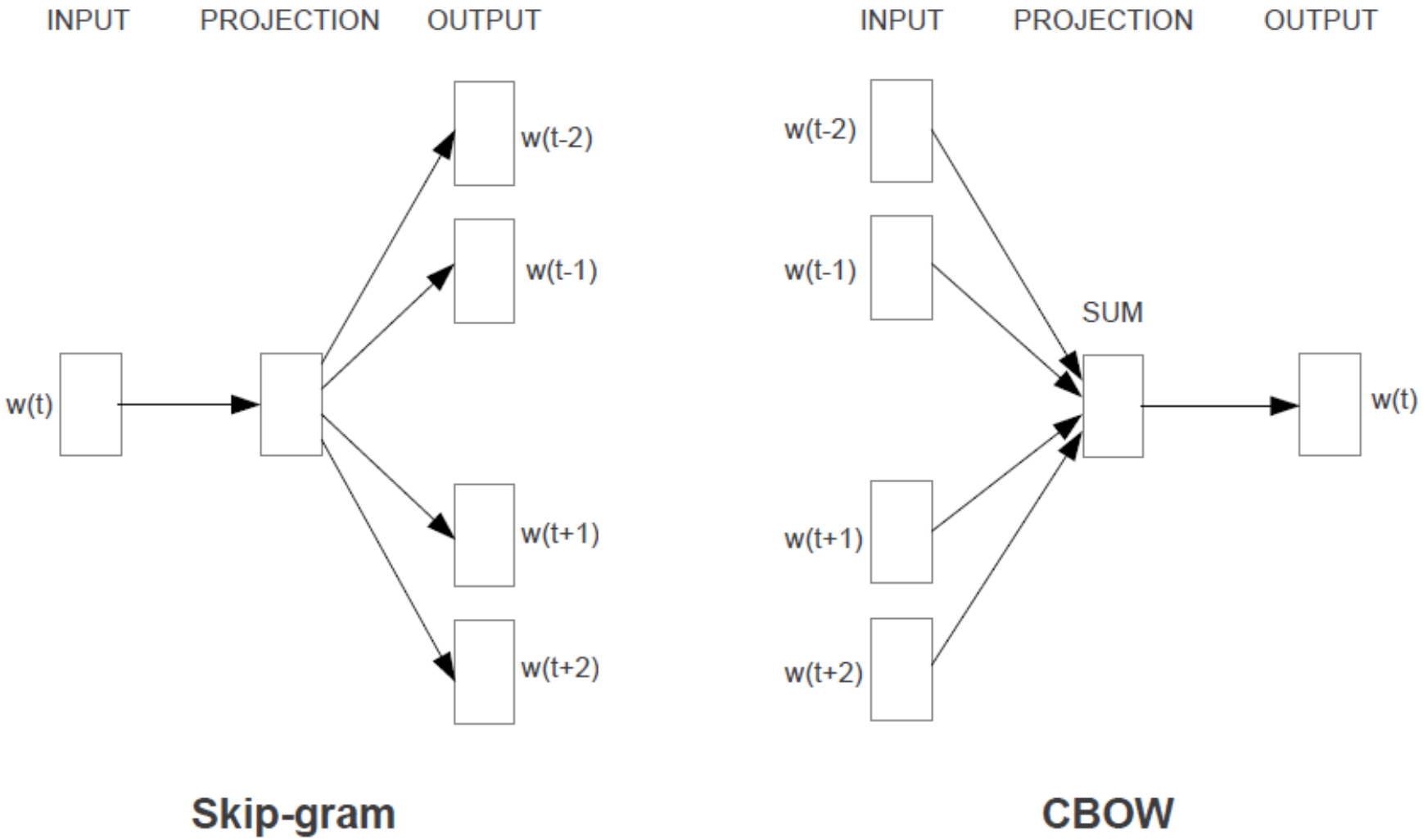


WORD2VEC

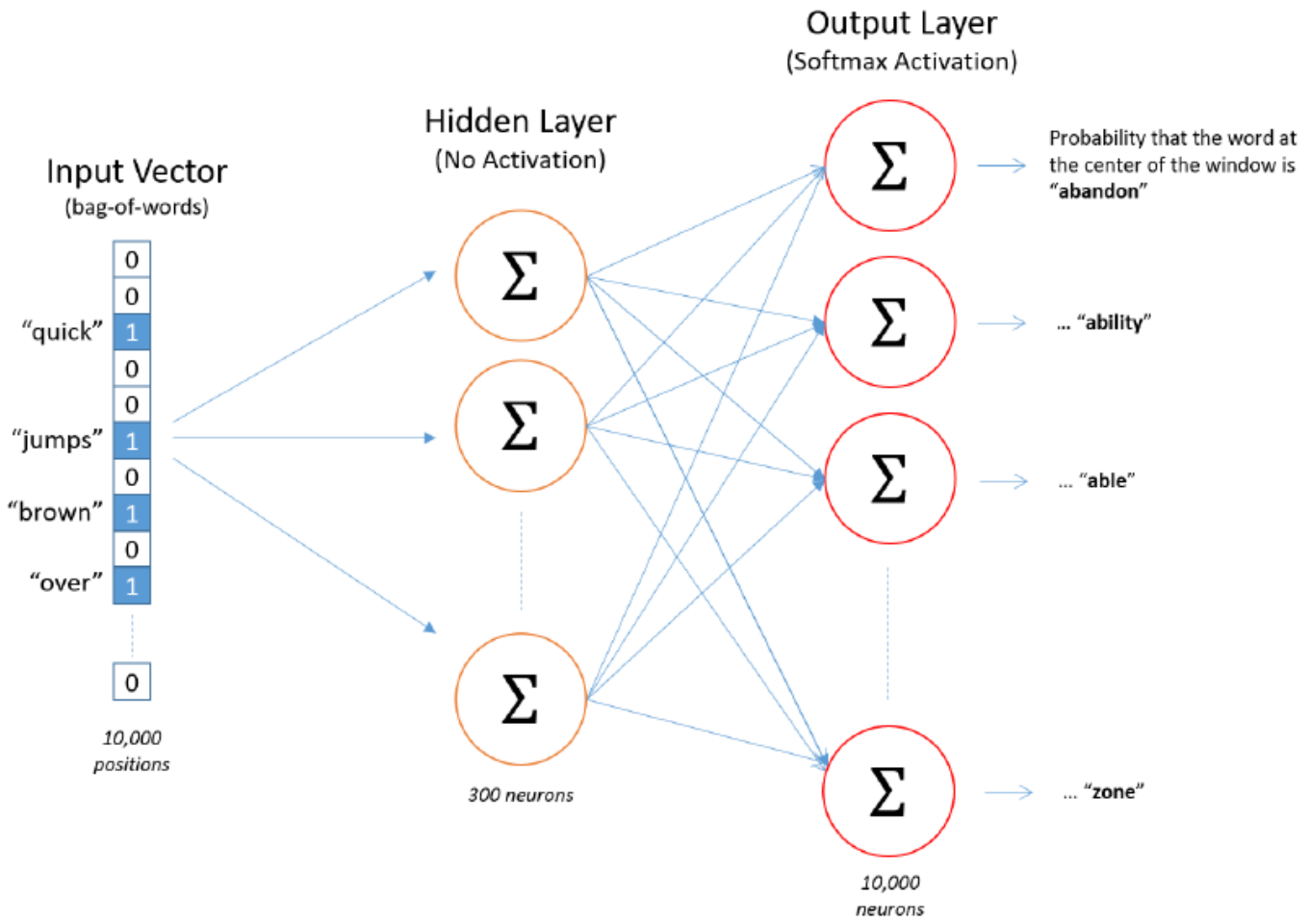
- Até agora, modelo *skip-gram*
  - Uma palavra prevê palavras de seu contexto

- Mas há também o *Continuous Bag-Of-Words* (CBOW)
  - O contexto é utilizado para prever uma palavra

Mikolov et al. (2013)



CONTINOUS BAG-OF-WORDS (CBOW)



WORD2VEC

- Camada escondida também chamada de "camada de projeção"
- Número de dimensões (tamanho das *embeddings*) é determinado empiricamente, normalmente, e depende da tarefa em vista
- Não há dominância entre skip-gram e cbow nas tarefas
  - Cada caso é um caso e precisa ser analisado

EXEMPLOS

- Mikolov et al., 2013

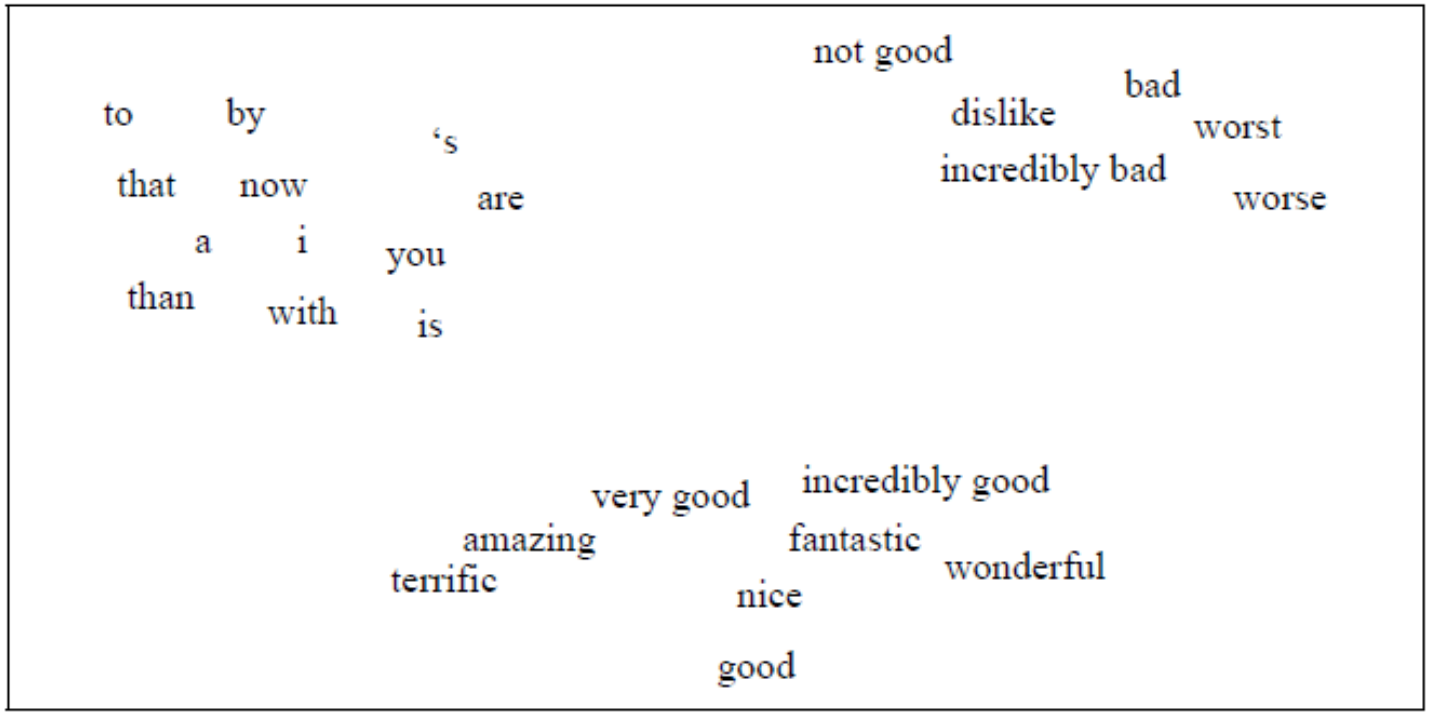
target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

Exemplos famosos

- vector('king') - vector('man') + vector('woman') ≈ vector('queen')
- vector('Paris') - vector('France') + vector('Italy') ≈ vector('Rome')
- vector('Germany') + vector('capital') ≈ vector('Berlin')

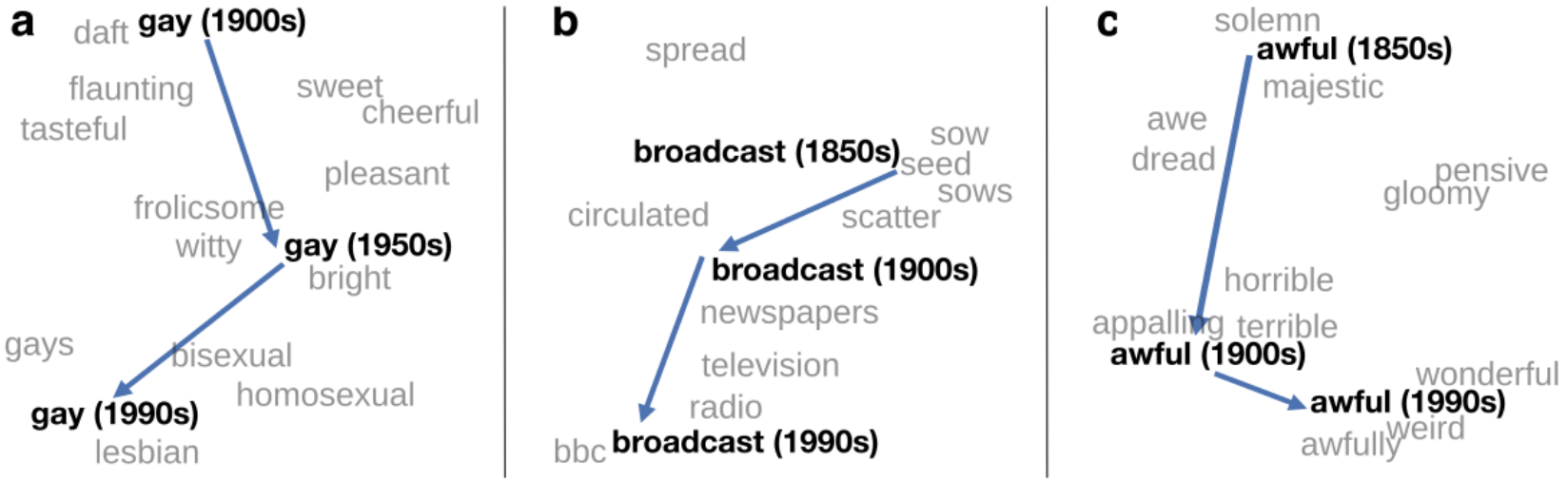
Exemplos

- Li et al., 2015
  - Espaço vetorial em uma tarefa de análise de sentimentos



Exemplos

- Mudanças históricas (Hamilton et al., 2016)



Exemplos

- Estereótipos e comportamentos sociais codificados nos vetores
  - Bolukbasi et al., (2016)
    - ‘computer programmer’ – ‘man’ + ‘woman’ = ?
    - ‘father’ → ‘doctor’ & ‘mother’ → ?

Exemplos

- Estereótipos e comportamentos sociais codificados nos vetores
  - Bolukbasi et al., (2016)
    - ‘computer programmer’ – ‘man’ + ‘woman’ = ‘homemaker’
    - ‘father’ → ‘doctor’ & ‘mother’ → ‘nurse’

Exemplos

- Estereótipos e comportamentos sociais codificados nos vetores
  - Caliskan et al. (2017)
    - Nomes americanos e europeus (‘Brad’, ‘Greg’, ‘Courtney’) relacionados a palavras boas
    - Nomes africanos (‘Leroy’ and ‘Shaniqua’) relacionados a palavras ruins

Prática

- Vamos checar se esses problemas se mantêm
  - WebVectors: http://vectors.nlpl.eu/explore/embeddings/en/

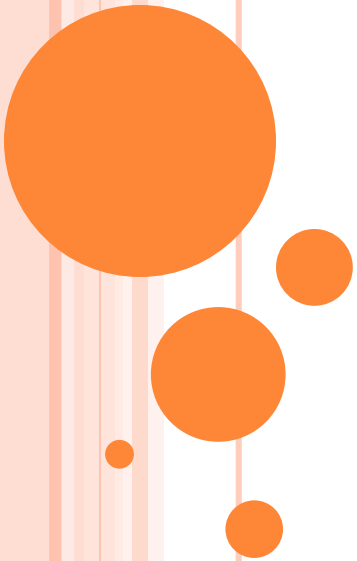
Fares et al. (2017)

Para testar

- *waiter* vs *waitress*
- *masculine* vs *feminine*
- *programmer* vs. *housekeeper*
- *Brazil* vs *Portugal*

Limitações de modelos à la word2vec?

- Não tem distinção para diferentes significados
  - "Manga" (fruta) e "manga" (de camisa) terão o mesmo vetor
- A semântica é implícita: não sabemos verdadeiramente qual o significado do termo
  - Temos apenas uma listagem de números
- Não há discriminação das relações diferentes que podem ocorrer entre os termos
- Só se aprende o que está nos dados
  - Para o bem e para o mal



BERT

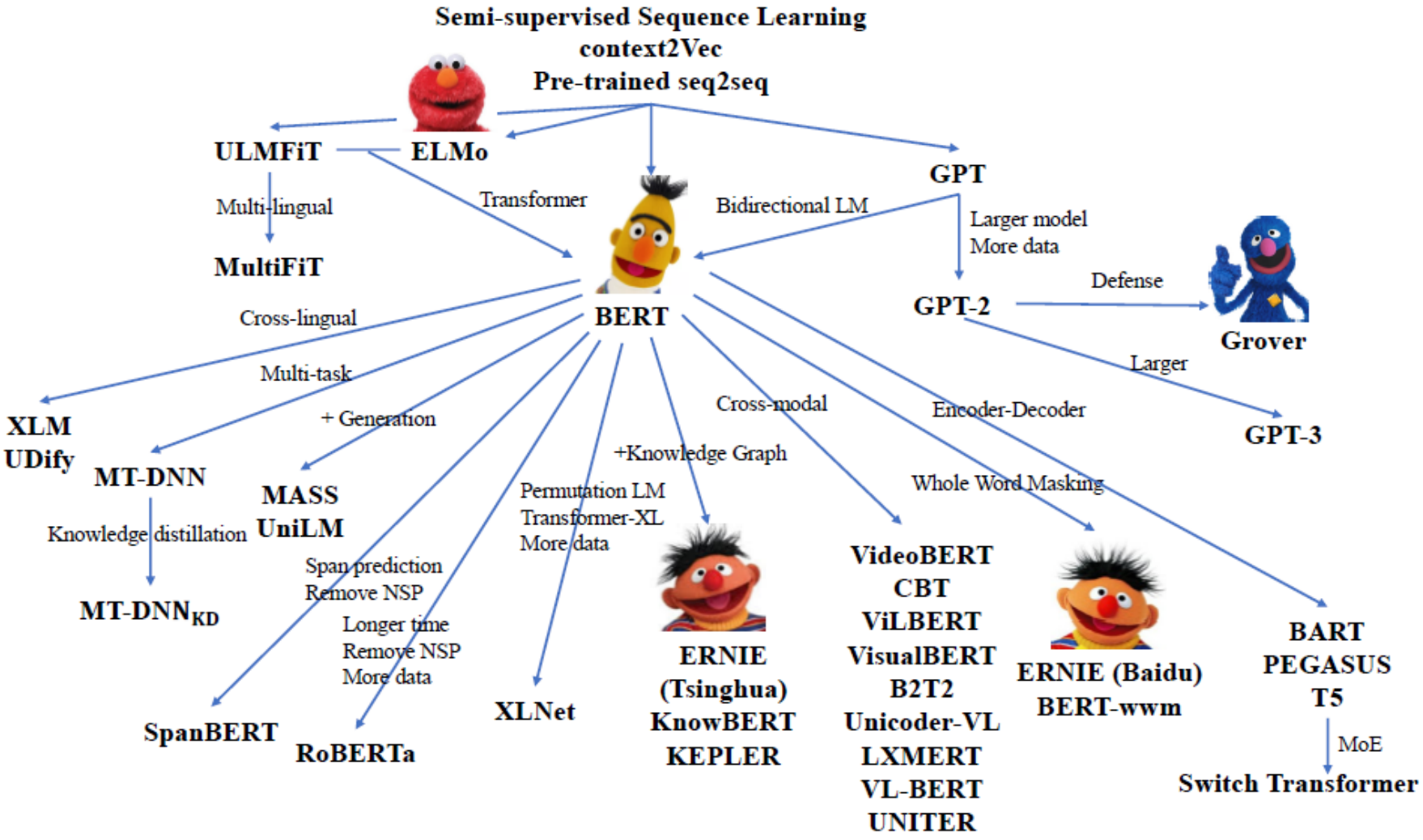
BERT

(Devlin et al., 2019)

- *Bidirectional Encoder Representations from Transformers*

- Diferencial
  - Solução "mais elegante" do que convoluções e recorrências
    - Atenção!
  - Computação paralelizável
    - Muito importante em função da quantidade de processamento e parâmetros envolvidos
  - *Embeddings* dinâmicos
  - Contexto levado em conta
    - O BERT é dito pertencer à família dos modelos "contextuais"

Pre-Trained Models: Past, Present and Future (Han et al., 2021)



BERT & TRANSFORMER

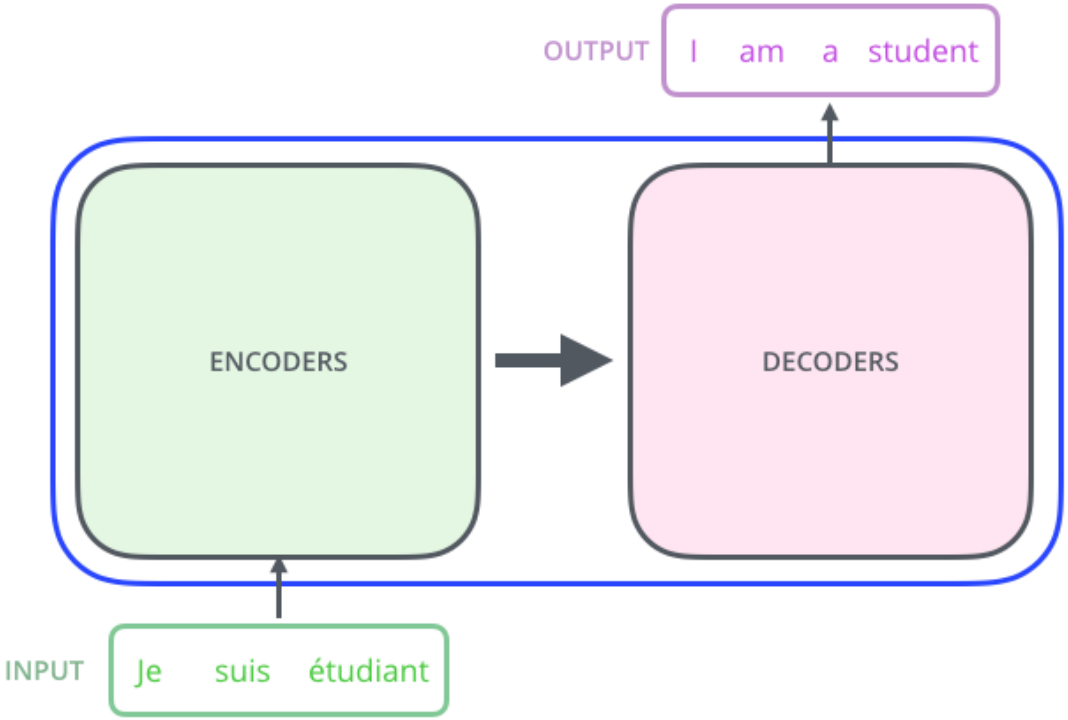
- Treinado com parte (modificada) da arquitetura do Transformer (Vaswani et al., 2017)
  - *Attention Is All You Need*

- Diversas tarefas de PLN produziram melhores resultados

- Indiretamente, criou uma área: "bertologia"
  - O que faz? Como faz? Por que faz?

Transformer

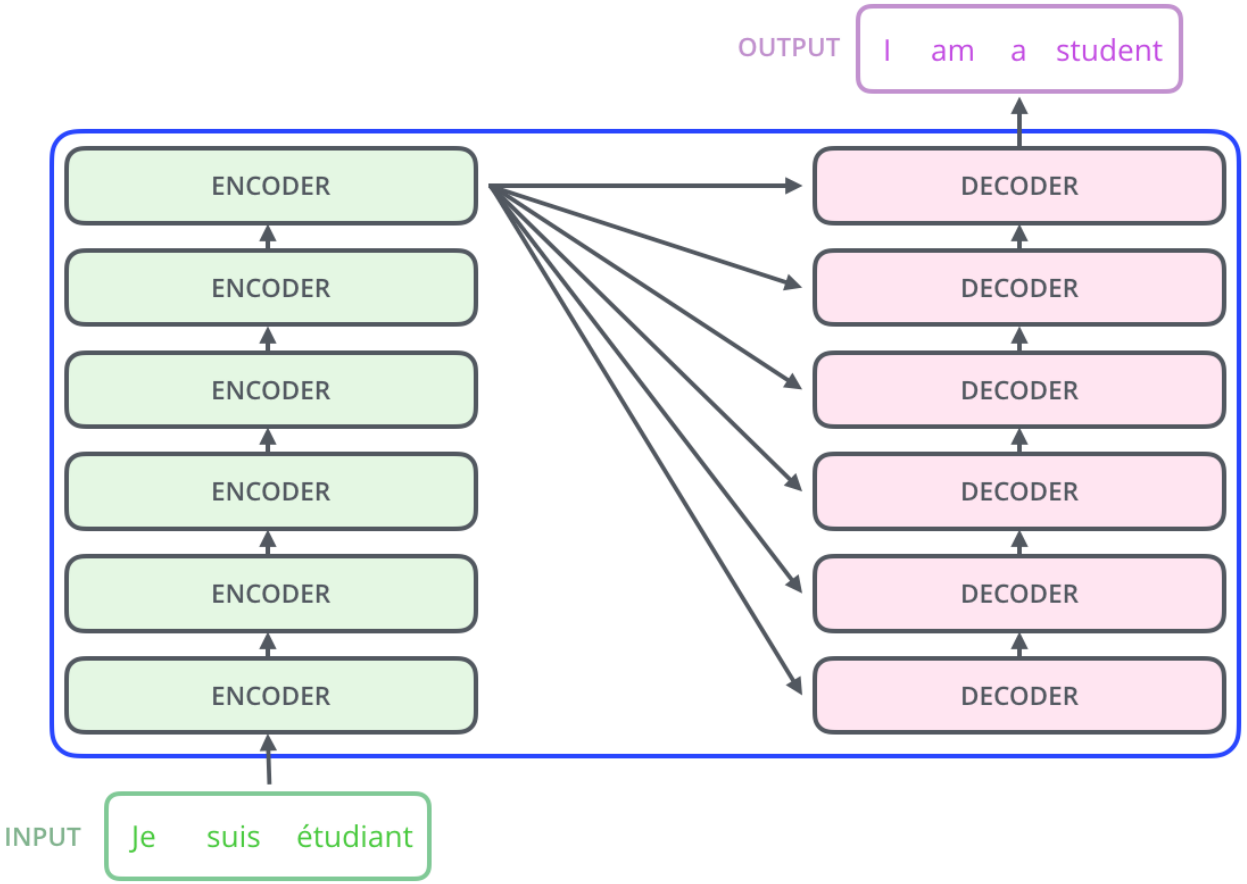
- Visão abrangente
  - Fez sua estreia na Tradução Automática



Fonte: *The Illustrated Transformer*

Transformer

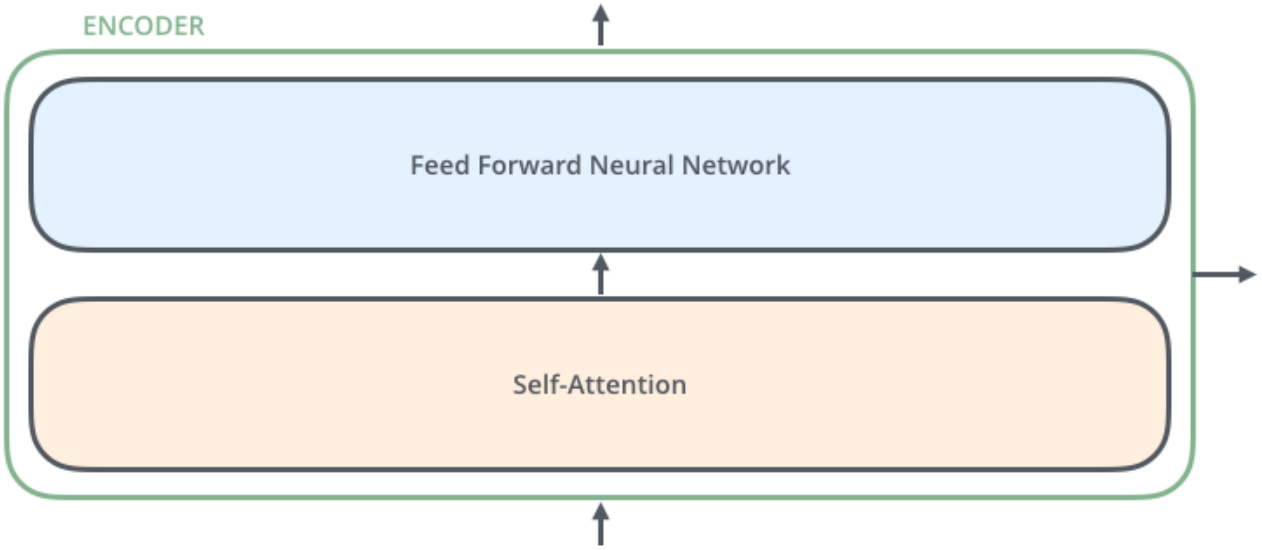
- Abrindo cada componente



Fonte: *The Illustrated Transformer*

Transformer

- Abrindo um “encoder”

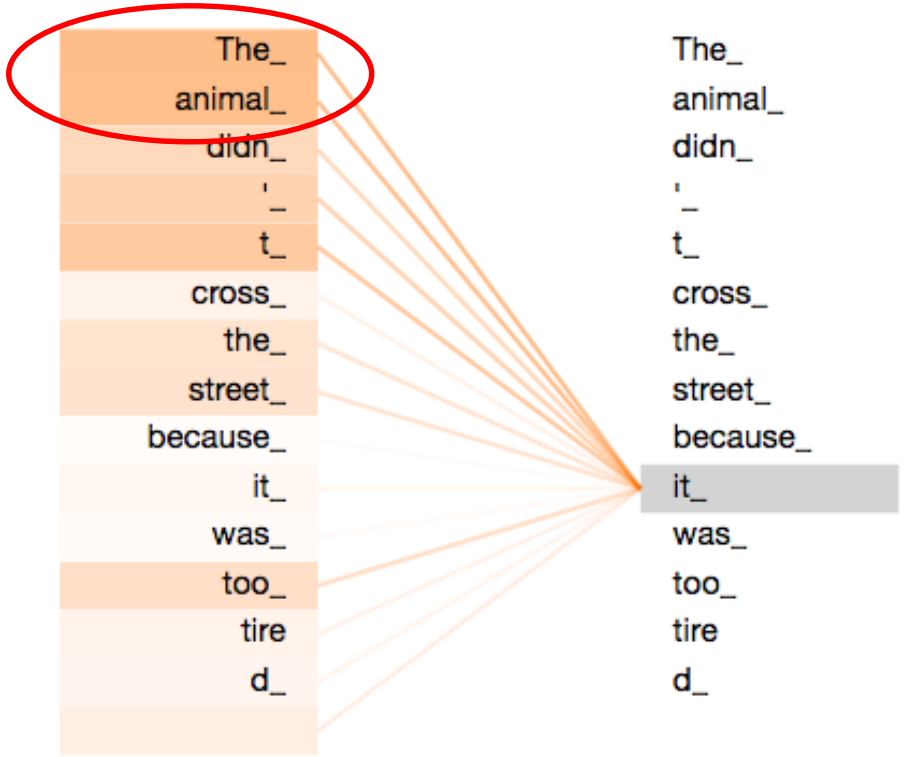


Fonte: *The Illustrated Transformer*

TRANSFORMER: ATENÇÃO

- Computando "atenção" e determinando o que é mais relevante para o processamento

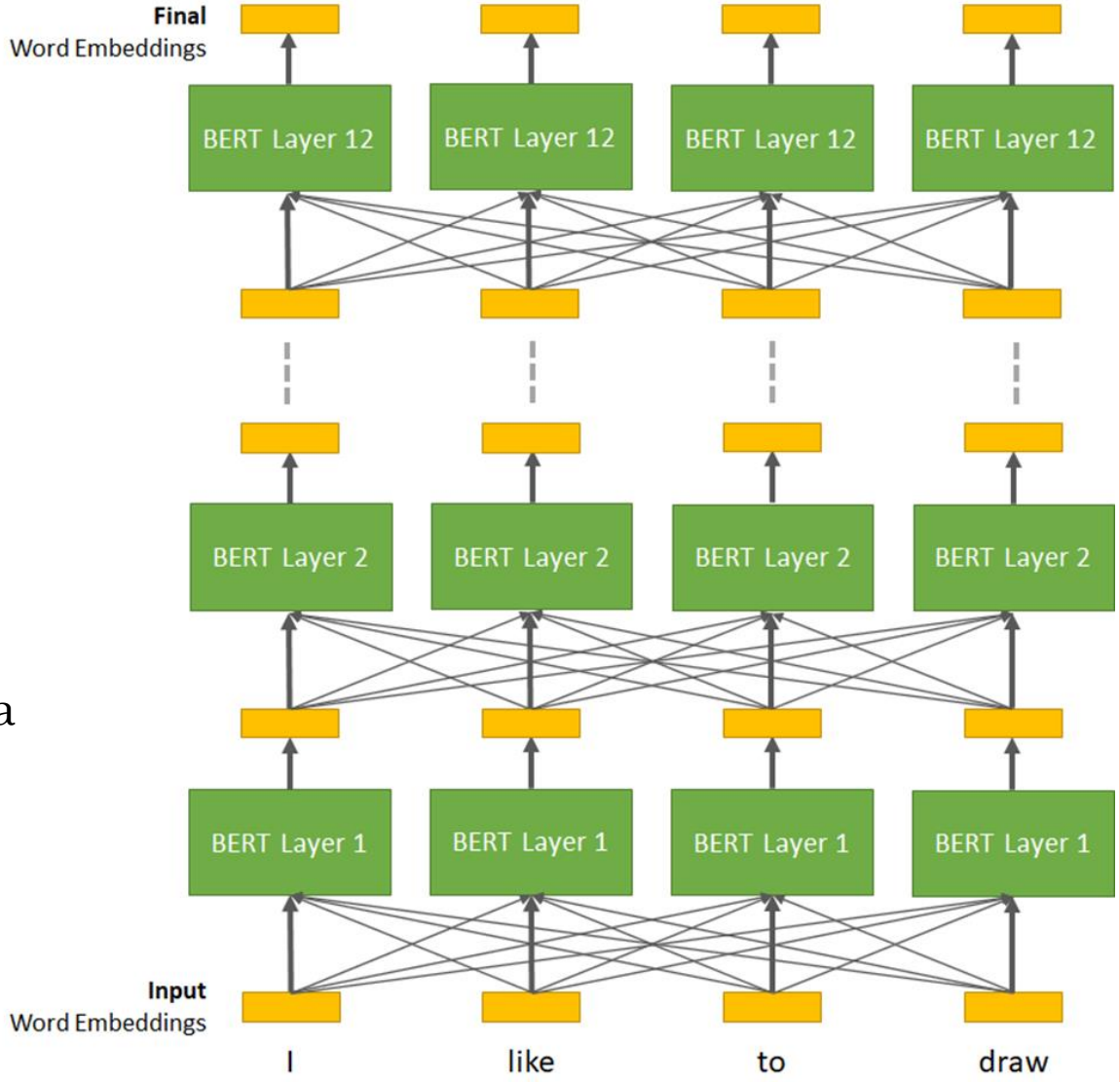
Procurando as palavras mais relevantes para o processamento de "it"



Fonte: *The Illustrated Transformer*

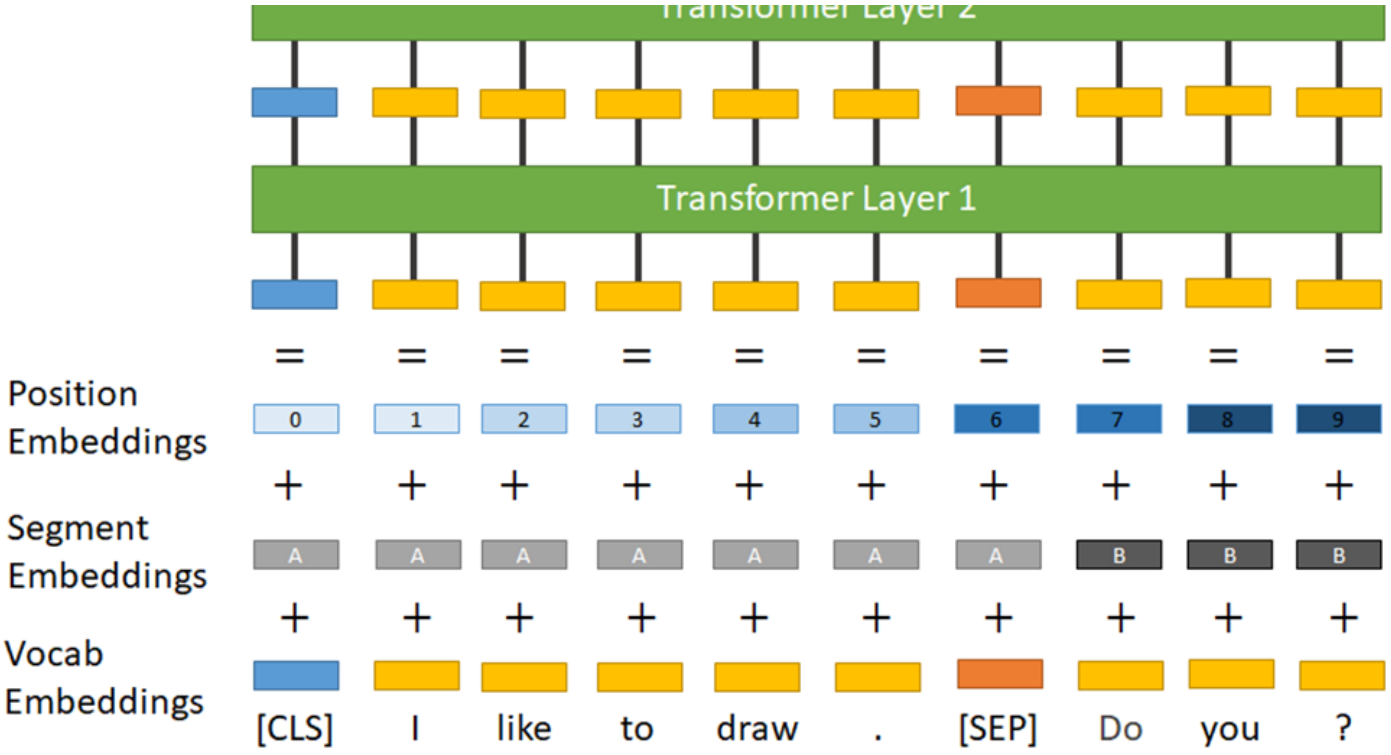
BERT

- 12 camadas de encoders na configuração mais comum
  - Cada elemento de cada camada pode ser executado em paralelo aos demais
  - Cada elemento recebe como entrada todas as palavras
    - E surge o contexto!



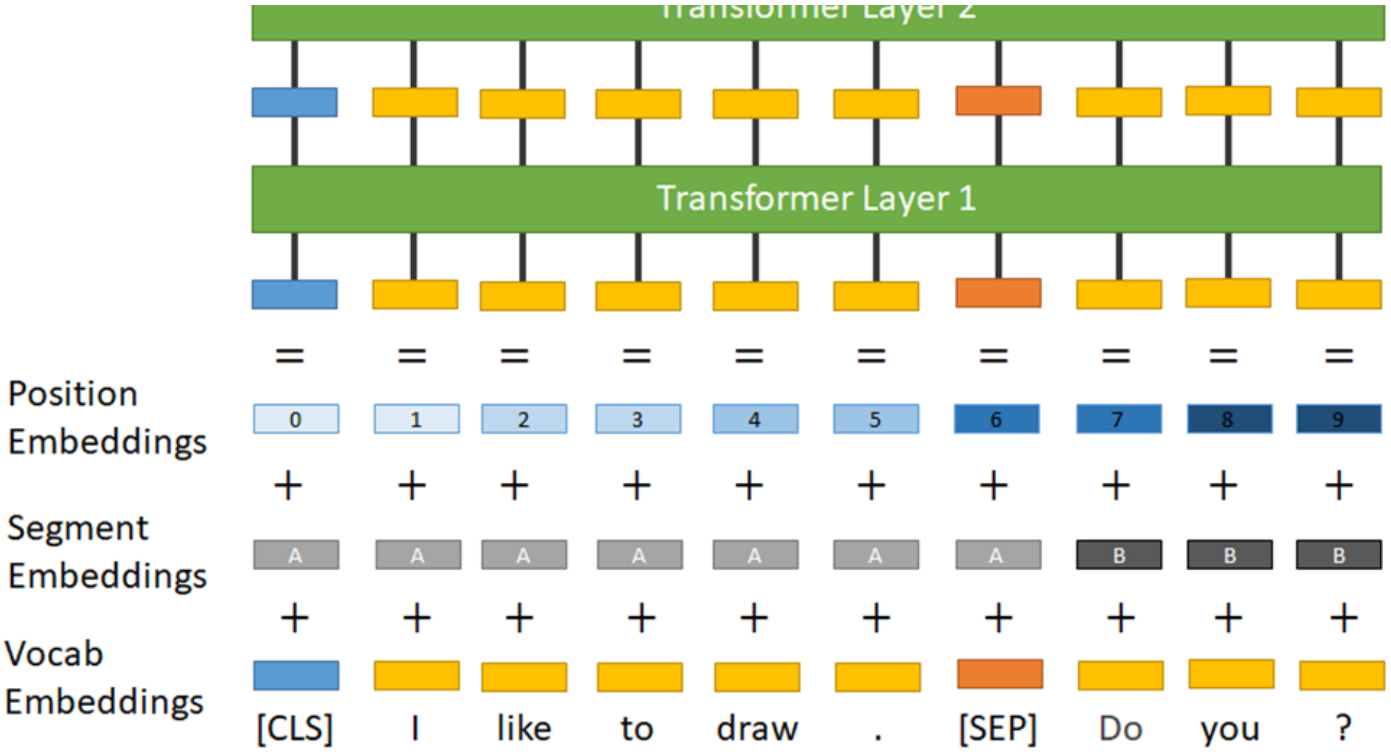
BERT

- Considerando pares de segmentos
  - Indicação dos segmentos das palavras
  - Tokens especiais: [CLS] e [SEP]



BERT

- Em mais detalhes
  - Tarefas "fake" utilizadas para treinamento
    - *Masked Language Model* (MLM)
    - *Next Sentence Prediction* (NSP)

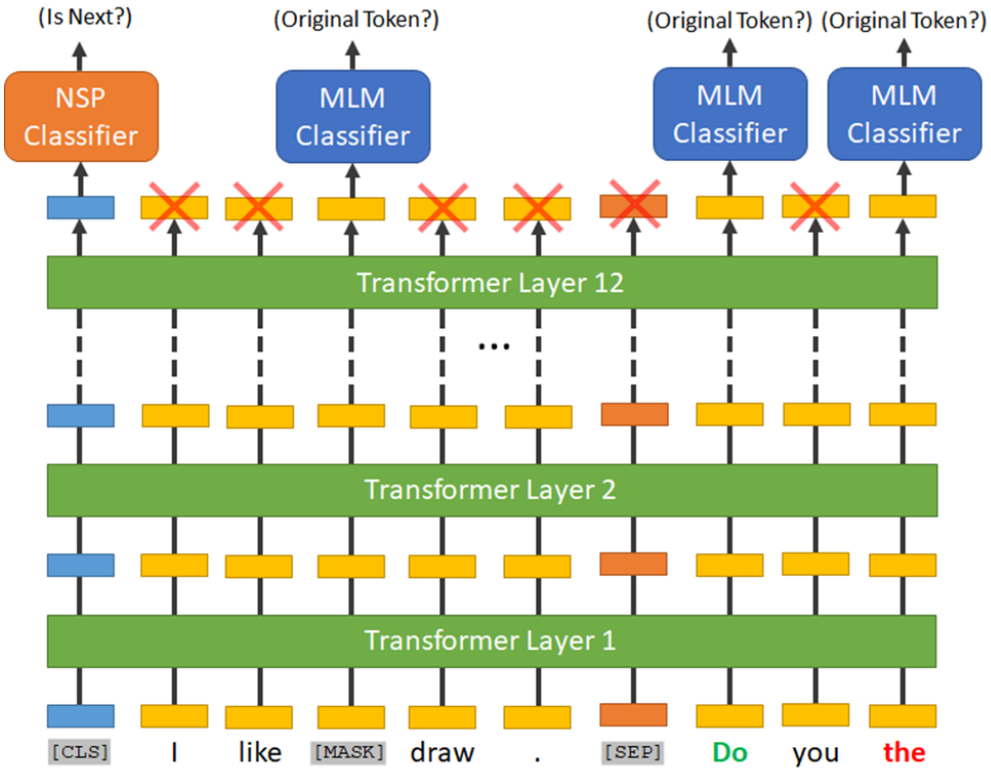


BERT

- Em mais detalhes
  - Tarefas "fake" utilizadas para treinamento
    - *Masked Language Model* (MLM)
    - *Next Sentence Prediction* (NSP)

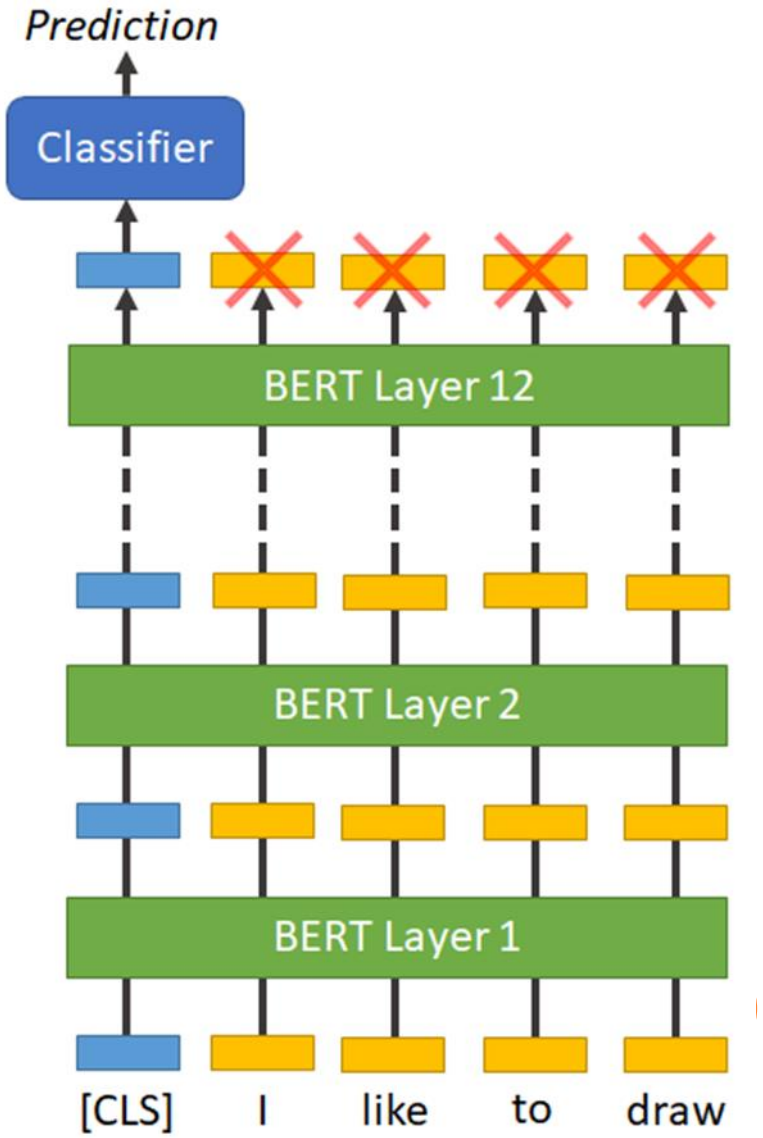
MLM: 15% dos tokens são "mascarados"

- 80% escondidos → [MASK]
- 10% trocados por token aleatório
- 10% mantidos



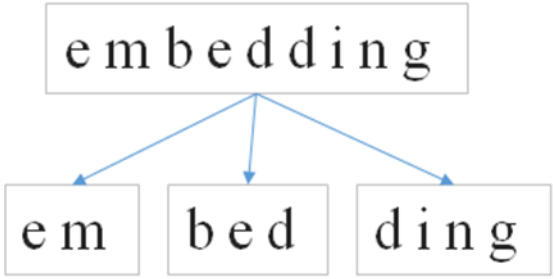
BERT

- *Embedding* do token [CLS] normalmente utilizado como *embedding* de todo o conteúdo apresentado, para diversas tarefas



Detalhes

- Tokenização: segmentos menores do que palavras
  - WordPiece (Wu et al., 2016)
    - Interessante para lidar com palavras fora do vocabulário e auxiliar na generalização do modelo

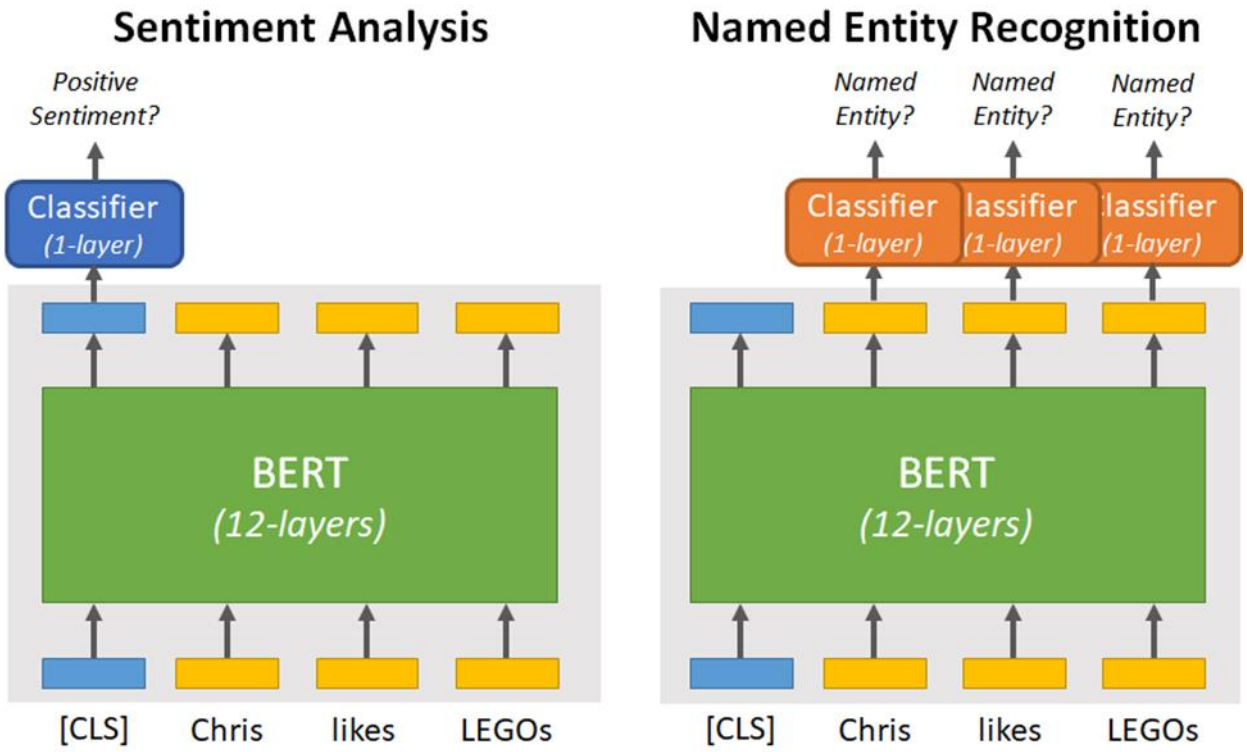


BERT e aprendizado

- As possibilidades de *transfer learning*
  - Etapas
    - *Pre-training* (normalmente pronto, utilizando-se as bases disponíveis)
    - *Fine-tuning*
      - Camadas adicionais para a tarefa de interesse
  - Vantagens
    - Desenvolvimento mais rápido e barato
    - Menos dados necessários
    - Resultados melhores, normalmente

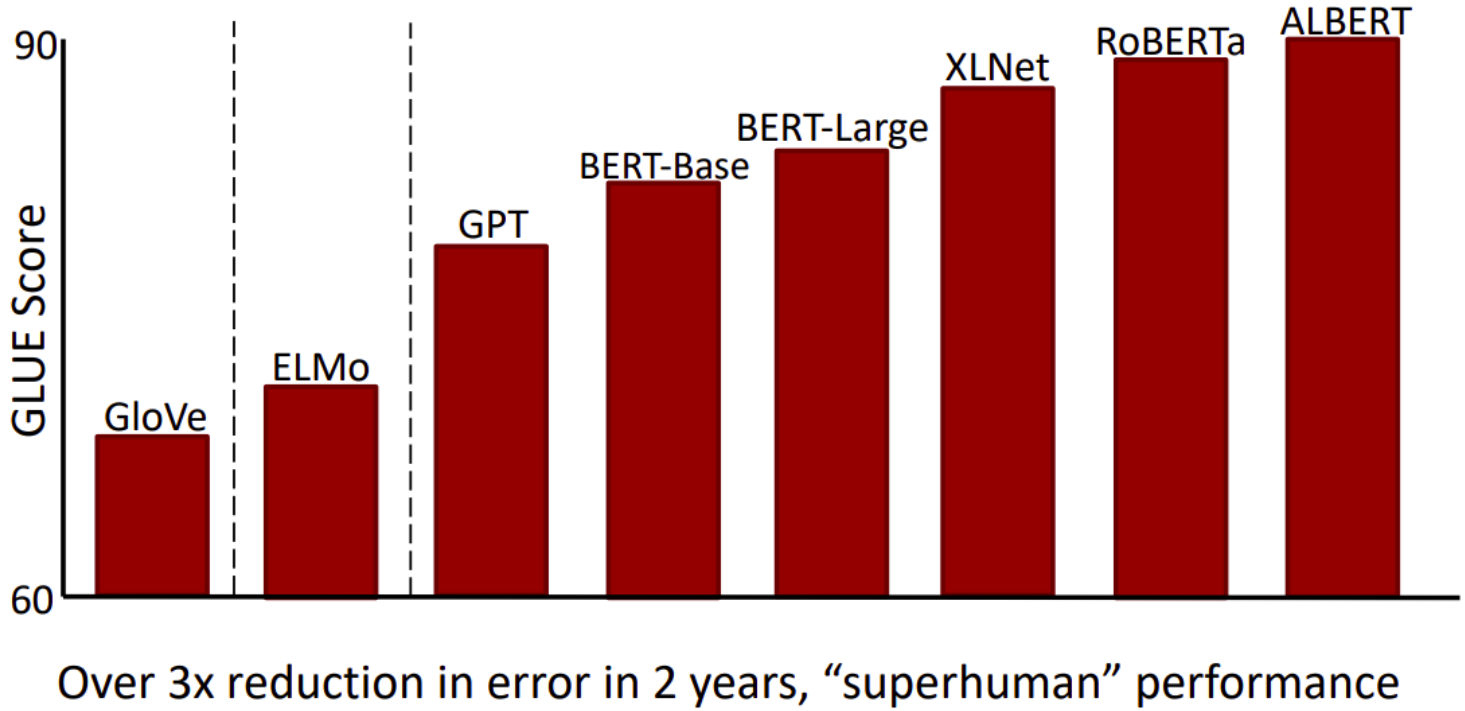
BERT E APRENDIZADO

- As possibilidades de *transfer learning*



Avanços significativos

- Ao longo do tempo: GLUE (*General Language Understanding Evaluation*) (Wang et al., 2019)



Limitações de modelos à la BERT

- A semântica ainda é implícita: não sabemos verdadeiramente qual o significado do termo
- Também não há discriminação das relações diferentes que podem ocorrer entre os termos
- Ainda só se aprende o que está nos dados
  - A semântica ainda é limitada!

Alguns desafios permanecem

```
>>> from transformers import pipeline
>>> unmasker = pipeline('fill-mask', model='bert-base-uncased')
>>> unmasker("The man worked as a [MASK].")

[{'sequence': '[CLS] the man worked as a carpenter. [SEP]',
  'score': 0.09747550636529922,
  'token': 10533,
  'token_str': 'carpenter'},
 {'sequence': '[CLS] the man worked as a waiter. [SEP]',
  'score': 0.0523831807076931,
  'token': 15610,
  'token_str': 'waiter'},
 {'sequence': '[CLS] the man worked as a barber. [SEP]',
  'score': 0.04962705448269844,
  'token': 13362,
  'token_str': 'barber'},
 {'sequence': '[CLS] the man worked as a mechanic. [SEP]',
  'score': 0.03788609802722931,
  'token': 15893,
  'token_str': 'mechanic'},
 {'sequence': '[CLS] the man worked as a salesman. [SEP]',
  'score': 0.037680890411138535,
  'token': 18968,
  'token_str': 'salesman'}]
```

Alguns desafios permanecem

```
>>> unmasker("The woman worked as a [MASK].")

[{'sequence': '[CLS] the woman worked as a nurse. [SEP]',
  'score': 0.21981462836265564,
  'token': 6821,
  'token_str': 'nurse'},
 {'sequence': '[CLS] the woman worked as a waitress. [SEP]',
  'score': 0.1597415804862976,
  'token': 13877,
  'token_str': 'waitress'},
 {'sequence': '[CLS] the woman worked as a maid. [SEP]',
  'score': 0.1154729500412941,
  'token': 10850,
  'token_str': 'maid'},
 {'sequence': '[CLS] the woman worked as a prostitute. [SEP]',
  'score': 0.037968918681144714,
  'token': 19215,
  'token_str': 'prostitute'},
 {'sequence': '[CLS] the woman worked as a cook. [SEP]',
  'score': 0.03042375110089779,
  'token': 5660,
  'token_str': 'cook'}]
```

O QUE O FUTURO NOS RESERVA?

- Nesta frente, é difícil prever
  - Há modelos especializados para tarefas particulares
  - Cada vez mais dados!
  - Há tentativas de "simplificação" dos modelos (menos parâmetros!) sem comprometer a qualidade geral
    - Limitações das plataformas computacionais utilizadas (smartphones, por exemplo)
    - Restrições da LGPD
  - A evolução tem sido rápida!
    - A área e o perfil dos trabalhos têm mudado radicalmente nos últimos anos
      - O que virá depois da "atenção"?

TRANSFORMERS EM MAIS DETALHES

- *The illustrated transformer*, por Jay Alammar

  - Um dos textos introdutórios mais claros e "gentis" sobre o tópico

Tarefa

- Leitura da semana
  - Bender, E.M. and Koller, A. (2020). Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data.
    - No e-Disciplinas