



Universidade de Brasília

Departamento de Ciência da Computação



Arquiteturas Encoder-Decoder

Curso ENAP - Processamento de Linguagem Natural

Prof. Dr. Vinícius Ruela Pereira Borges

`viniciusrpb@unb.br`

Brasília-DF, 2024

- Esses slides foram redigidos e produzidos pelo Prof. Dr. Vinícius R. P. Borges;
- Material didático de referência:
 - Capítulo 10 do livro “Speech and Language Processing. Daniel Jurafsky & James H. Martin, 2021.”
 - Slides do curso “CS224n: Natural Language Processing with Deep Learning” - Stanford University

- Artigos científicos:
 - Sutskever, I., Vinyals, O., Le, Q. V.. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27, 2014.
 - Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.
 - Nallapati, R., Xiang, B., Zhou, B.. Sequence-to-sequence rnns for text summarization, Workshop of the International Conference on Learning Representations, 2016.

- Motivação

- Motivação
- LSTM Bidirecional

- Motivação
- LSTM Bidirecional
- Tradução automática

- Motivação
- LSTM Bidirecional
- Tradução automática
- Seq2Seq

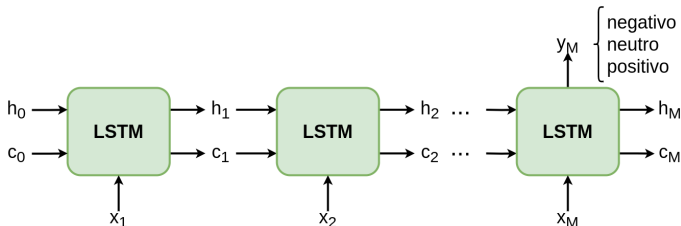
- Motivação
- LSTM Bidirecional
- Tradução automática
- Seq2Seq
- Considerações Finais

Motivação

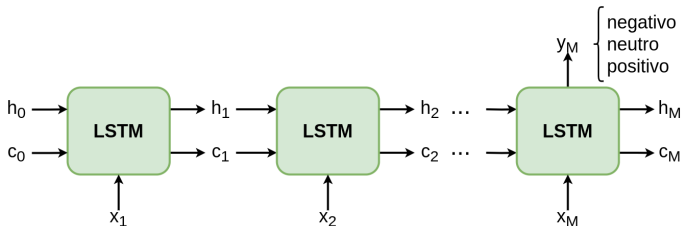


- Em problemas de classificação de texto, vimos que LSTMs podem codificar uma sentença;
- O que significa codificar uma sentença?
 - Gerar uma codificação que caracteriza os padrões implícitos de uma sentença;
 - Contexto da sentença!

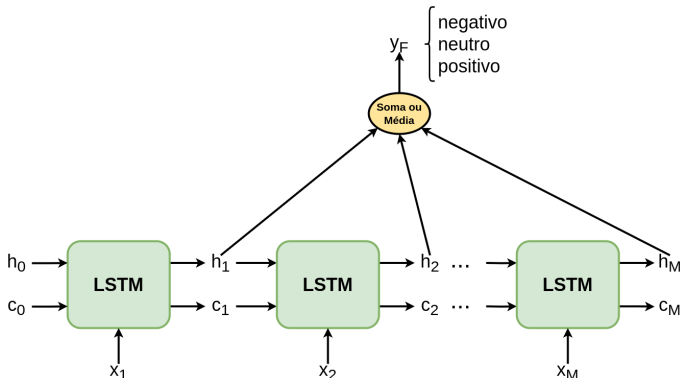
- Se processarmos uma sentença composta pelas palavras cujos vetores são x_1, x_2, \dots, x_M :



- Basta pegar o último estado h_M como a **codificação da sentença**:

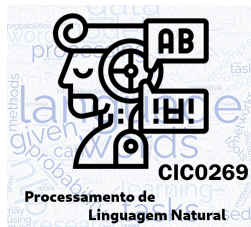


- ... ou então fazer uma soma / média de todas as h_i como a **codificação da sentença**:



- Pode-se fazer adaptações na codificação da sentença para se capturar mais informações das sentenças de entrada!

LSTM Bidirecional



- Consiste em utilizar duas células LSTM distintas (parâmetros não são compartilhados), uma para cada ordem de percorrimento da sentença de entrada

- Consiste em utilizar duas células LSTM distintas (parâmetros não são compartilhados), uma para cada ordem de percorrimento da sentença de entrada
 - Da esquerda para direita;
 - Da direita ou para esquerda.

- Consiste em utilizar duas células LSTM distintas (parâmetros não são compartilhados), uma para cada ordem de percorrimento da sentença de entrada
 - Da esquerda para direita;
 - Da direita ou para esquerda.
- A ideia é capturar o contexto nos dois sentidos e armazená-los em uma única representação vetorial.

- A célula LSTM que processa as palavras da sentença de entrada da esquerda para a direita é dada por:

$$h_F^{(t)} = LSTM_F(h^{(t-1)}, c^{(t-1)}, x^{(t)}) \quad (1)$$

- A célula LSTM que processa as palavras da sentença de entrada da esquerda para a direita é dada por:

$$h_F^{(t)} = LSTM_F(h^{(t-1)}, c^{(t-1)}, x^{(t)}) \quad (2)$$

- Já a célula LSTM que faz o processamento da direita para a esquerda é descrita por:

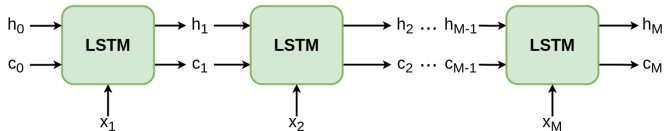
$$h_B^{(t)} = LSTM_B(h^{(t+1)}, c^{(t+1)}, x^{(t)}) \quad (3)$$

- Por fim, concatena-se os dois vetores dos estados internos:

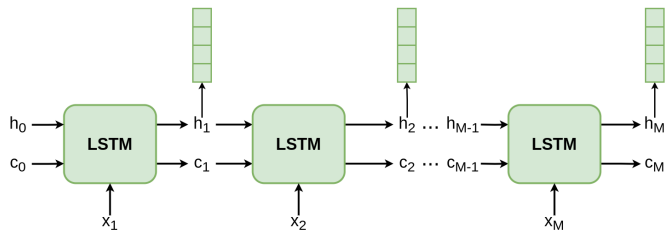
$$h^{(t)} = [h_F^{(t)}, h_B^{(t)}] \quad (4)$$

- Este é o estado interno da LSTM Bidirecional!

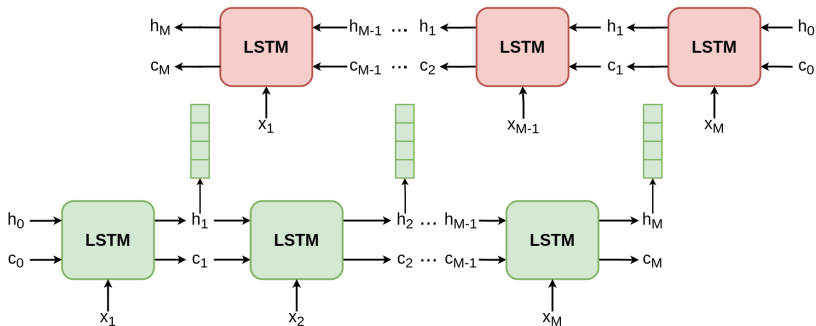
LSTM: Sentido Forward



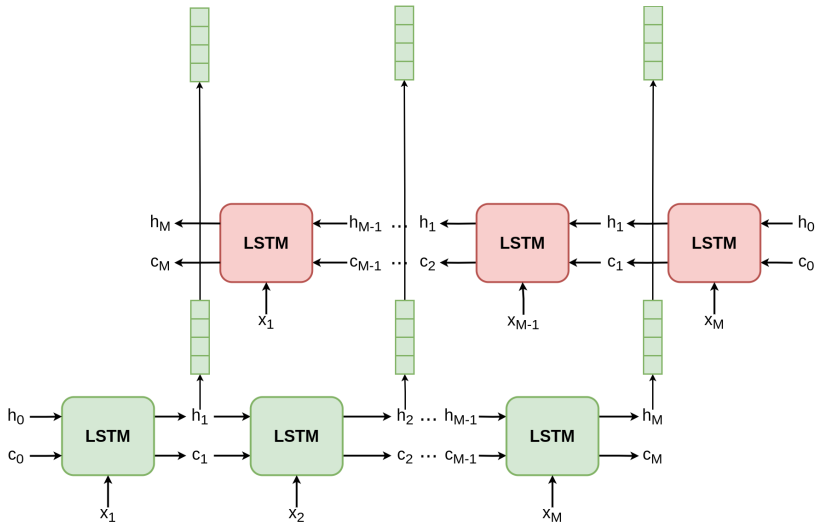
LSTM: Sentido Forward



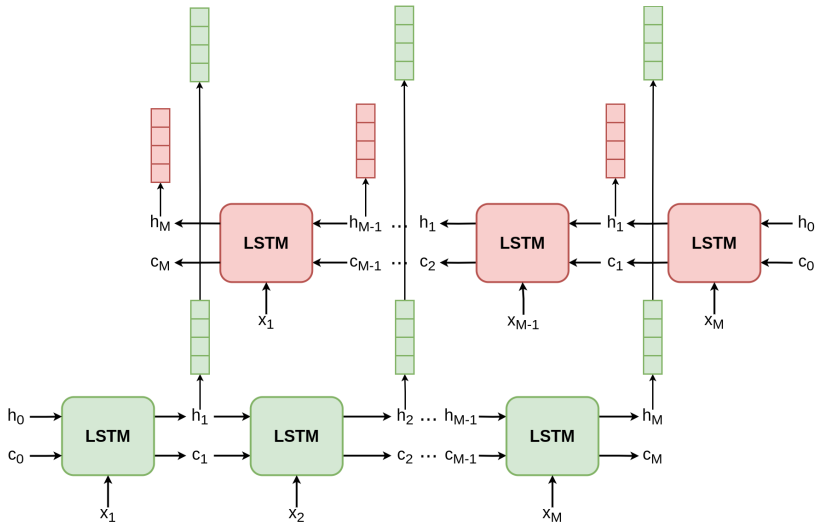
LSTM: Sentido Backward



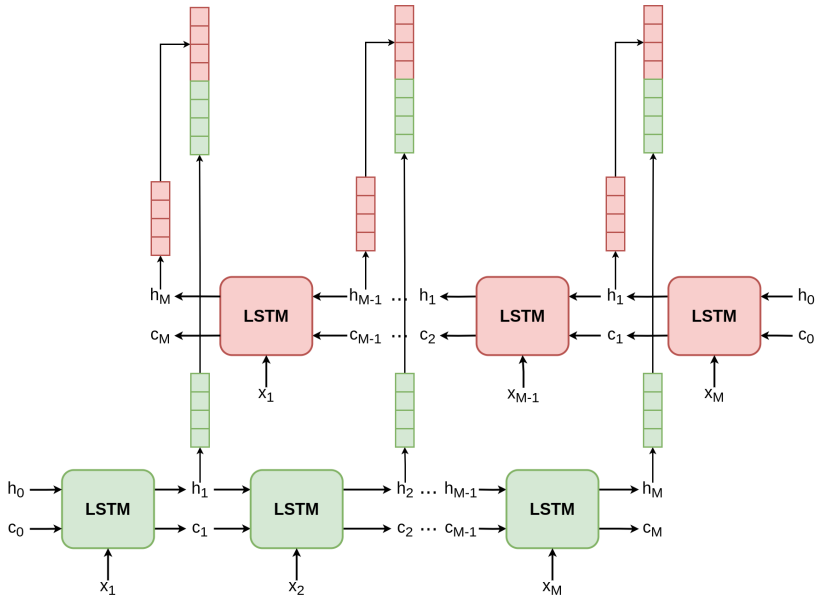
LSTM: Sentido Backward



LSTM: Sentido Backward



LSTM: Sentido Forward



- BiLSTMs são aplicáveis apenas quando se tem acesso à sequência de entrada completa;

- BiLSTMs são aplicáveis apenas quando se tem acesso à sequência de entrada completa;
 - não são aplicáveis em modelos de linguagem, pois estes models só conhecem as palavras anteriores da esquerda!

- BiLSTMs são aplicáveis apenas quando se tem acesso à sequência de entrada completa;
 - não são aplicáveis em modelos de linguagem, pois estes modelos só conhecem as palavras anteriores da esquerda!
- Se você possui a sentença completa disponível, compensa investir em uma abordagem bidirecional.
 - Cenas dos próximos capítulos: Bidirectional Encoder Representations from Transformers (BERT)!

Tradução Automática



- Tradução automática (*Machine translation*) é a tarefa de traduzir uma sentença S_O de uma linguagem de origem para uma sentença S_D em uma linguagem de destino;

- Entre 1990 e 2010, a abordagem estatística era a mais comum de ser implementada;

- Entre 1990 e 2010, a abordagem estatística era a mais comum de ser implementada;
- A ideia é aprender um modelo probabilístico sobre os textos;

- Entre 1990 e 2010, a abordagem estatística era a mais comum de ser implementada;
- A ideia é aprender um modelo probabilístico sobre os textos;
- Suponha que queiramos encontrar uma sentença S_D em Inglês a partir de uma sentença S_O escrita em Francês;

- Suponha que queiramos encontrar a melhor sentença S_D em Inglês possível a partir de uma sentença S_O escrita em Francês;

$$\arg \max_{S_D} P(S_D|S_O) \quad (5)$$

- Suponha que queiramos encontrar a melhor sentença S_D em Inglês possível a partir de uma sentença S_O escrita em Francês;

$$\arg \max_{S_D} P(S_D|S_O) \quad (6)$$

- Utiliza-se a regra de Bayes para separar dois componentes que serão determinados separadamente:

$$\arg \max_{S_D} P(S_O|S_D)P(S_D) \quad (7)$$

$$\arg \max_{S_D} P(S_O | S_D) P(S_D) \quad (8)$$

$$\arg \max_{S_D} P(S_O|S_D)P(S_D) \quad (9)$$

- $P(S_O|S_D)$ se trata do modelo de tradução, isto é, modela como as palavras e as frases devem ser traduzidas;

$$\arg \max_{S_D} P(S_O|S_D)P(S_D) \quad (10)$$

- $P(S_O|S_D)$ se trata do modelo de tradução, isto é, modela como as palavras e as frases devem ser traduzidas;
 - Necessitam de informações externas para o aprendizado

$$\arg \max_{S_D} P(S_O|S_D)P(S_D) \quad (11)$$

- $P(S_O|S_D)$ se trata do modelo de tradução, isto é, modela como as palavras e as frases devem ser traduzidas;
 - Necessitam de informações externas para o aprendizado
- $P(S_D)$ modela como escrever um bom inglês!

$$\arg \max_{S_D} P(S_O|S_D)P(S_D) \quad (12)$$

- $P(S_O|S_D)$ se trata do modelo de tradução, isto é, modela como as palavras e as frases devem ser traduzidas;
 - Necessitam de informações externas para o aprendizado
- $P(S_D)$ modela como escrever um bom inglês!
 - Aprendido de textos em língua Inglesa.

- Os melhores tradutores automáticos eram extremamente complexos;

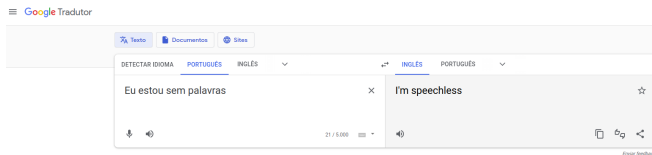
- Os melhores tradutores automáticos eram extremamente complexos;
- Muita engenharia de dados!

- Os melhores tradutores automáticos eram extremamente complexos;
- Muita engenharia de dados!
 - Uso de dicionários;

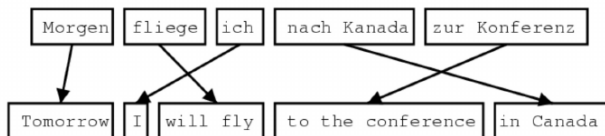
- Os melhores tradutores automáticos eram extremamente complexos;
- Muita engenharia de dados!
 - Uso de dicionários;
 - Criação de atributos para capturar detalhes particulares da linguagem;

- Os melhores tradutores automáticos eram extremamente complexos;
- Muita engenharia de dados!
 - Uso de dicionários;
 - Criação de atributos para capturar detalhes particulares da linguagem;
 - Tabelas contendo exemplos de frases equivalentes;

- Os melhores tradutores automáticos eram extremamente complexos;
- Muita engenharia de dados!
 - Uso de dicionários;
 - Criação de atributos para capturar detalhes particulares da linguagem;
 - Tabelas contendo exemplos de frases equivalentes;
- Repetia-se o esforço para cada par de linguagem.



- Repare que não basta fazer uma tradução literal palavra por palavra;



- Regras gramaticais diferentes entre as linguagens!

- Também chamada de *Neural Machine Translation*, consiste em empregar uma rede neural para a tarefa de tradução automática;

- Também chamada de *Neural Machine Translation*, consiste em empregar uma rede neural para a tarefa de tradução automática;
- Entretanto, um único modelo de linguagem, como uma Long Short-Term Memory (LSTM), não é suficiente...

- Também chamada de *Neural Machine Translation*, consiste em empregar uma rede neural para a tarefa de tradução automática;
- Entretanto, um único modelo de linguagem, como uma Long Short-Term Memory (LSTM), não é suficiente...
- Solução: considerar duas redes neurais recorrentes distintas.

- Objetivo de um modelo de linguagem é prever uma palavra ou caractere...

- Objetivo de um modelo de linguagem é prever uma palavra ou caractere...
- **Desafio:** o tamanho do texto colocado como entrada é diferente do tamanho do texto gerado na saída.

- Objetivo de um modelo de linguagem é prever uma palavra ou caractere...
- **Desafio: o tamanho do texto colocado como entrada é diferente do tamanho do texto gerado na saída.**
- Temos um problema de gerar uma sequência a partir de uma outra sequência fornecida da entrada!

Seq2Seq



- Modelos *sequence to sequence* (*seq2seq*) são arquiteturas capazes de gerar um texto a partir de outro texto, em que os tamanhos desses textos podem ser diferentes;

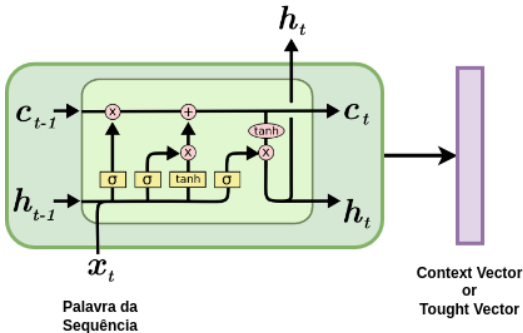
- Modelos *sequence to sequence* (*seq2seq*) são arquiteturas capazes de gerar um texto a partir de outro texto, em que os tamanhos desses textos podem ser diferentes;
- Esses modelos se baseiam uma arquitetura conhecida como Encoder-Decoder e são utilizadas para diferentes tipos de tarefas de modelagem de sequências.

- Visão geral de uma arquitetura encoder-decoder:



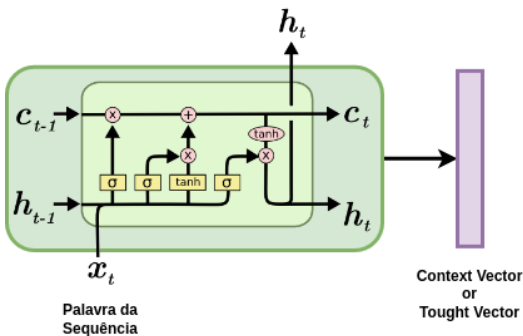
Encoder

- Recebe uma sequência em linguagem natural e converte para uma representação estruturada densa de baixa dimensão;



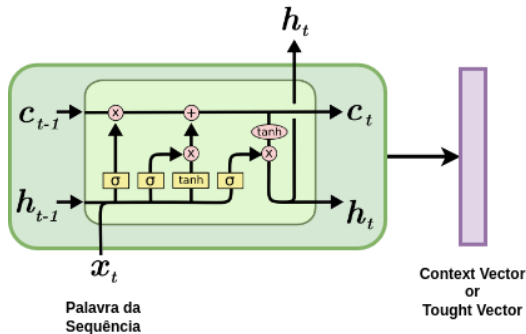
Encoder

- Tal representação pode ser denotada por um vetor numérico (codificação), denominado *Context Vector* ou *Dense Vector*.

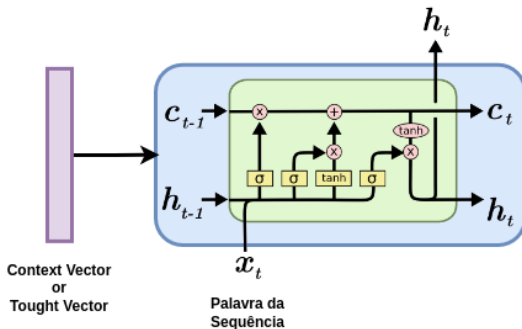


Encoder

- Observe que no caso de uma LSTM, o contexto de uma sequência de entrada pode ser extraídos dos seus estados internos:
 - Estados ocultos (*hidden states*);
 - Estados da célula (*cell states*);



- Gera uma sequência em linguagem natural a partir de uma codificação (vetor numérico);



- Gera uma sequência em linguagem natural a partir de uma codificação (vetor numérico);
- Um modelo de linguagem é capaz de gerar sequências de palavras...
- A qualidade das sequências geradas depende do contexto e dos embeddings!

Implementação



- O Encoder e o Decoder são treinados juntamente visando a geração da sequência de saída;
- Três partes são fundamentais:
 - Sequência de Entrada do Encoder;
 - Sequência de Entrada do Decoder;
 - Sequência de Saída do Decoder.

- Sequência de Entrada do Encoder: sequência de palavras na língua de origem;
- Sequência de Entrada do Decoder: uma palavra inicial;
- Sequência de Saída do Decoder: sequência de palavras correspondente na língua alvo;

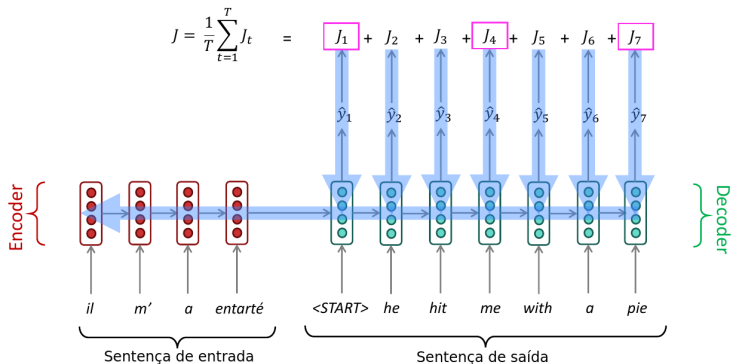
Implementação do Decoder

- No caso do Decoder, o treinamento é feito por uma estratégia conhecida como *teacher forcing*;
 - Fornece o estado inicial (*context vector*) do Encoder;
 - Treina o Decoder para produzir as sequências esperadas a partir da entrada e tentar prever a mesma entrada (um modelo de linguagem!).
- Por isso, as sequências de entrada e saída do Decoder são as mesmas, com a diferença de um *offset* de tempo.

Implementação do Encoder

- No seq2seq, não se pode treinar o Encoder de maneira isolada, uma vez que não há um *context vector* alvo a ser aprendido para fazer previsões;
- O backpropagation que vai atualizar os parâmetros do Encoder utiliza os gradientes (erros) provenientes do Decoder;
- Os estados internos (*hidden* e *cell* states) constituem o *context vector*.

Backpropagation



- Preferencialmente um grande *corpus* paralelo;

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino
- Criação de um vocabulário com palavras especiais:

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino
- Criação de um vocabulário com palavras especiais:
 - ‘<sos>’: início de sentença

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino
- Criação de um vocabulário com palavras especiais:
 - ‘<sos>’: início de sentença
 - ‘<eos>’: fim de sentença

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino
- Criação de um vocabulário com palavras especiais:
 - ‘<sos>’: início de sentença
 - ‘<eos>’: fim de sentença
 - ‘<unk>’: palavra desconhecida

- Preferencialmente um grande *corpus* paralelo;
 - Sentença na língua de origem → sentença na língua de destino
- Criação de um vocabulário com palavras especiais:
 - ‘<sos>’: início de sentença
 - ‘<eos>’: fim de sentença
 - ‘<unk>’: palavra desconhecida
 - ‘<pad>’: padding, para ajuste do tamanho de sentenças

- Por exemplo: se traduzirmos a sentença
“Eu estou sem palavras”

- Por exemplo: se traduzirmos a sentença
“Eu estou sem palavras”
- para a sentença destino:
“I am speechless”

- Por exemplo: se traduzirmos a sentença

“Eu estou sem palavras”

- para a sentença destino:

“I am speechless”

- [“<sos>”, “eu”, “estou”, “sem”, “palavras”, “<eos>”]
- [“<sos>”, “i”, “am”, “speechless”, “<pad>”, “<eos>”]

Considerações Finais



- A arquitetura Encoder-Decoder utilizada em modelos Seq2Seq é capaz de lidar com textos de entrada e saída de diferentes tamanhos;

- A arquitetura Encoder-Decoder utilizada em modelos Seq2Seq é capaz de lidar com textos de entrada e saída de diferentes tamanhos;
- Os modelos seq2seq codifica uma sentença de entrada para depois modelar uma sentença de saída a partir da codificação obtida;

- A arquitetura Encoder-Decoder utilizada em modelos Seq2Seq é capaz de lidar com textos de entrada e saída de diferentes tamanhos;
- Os modelos seq2seq codifica uma sentença de entrada para depois modelar uma sentença de saída a partir da codificação obtida;
- Boa capacidade de generalização, devido à codificação da sentença.

- Um modelo seq2seq processa toda a entrada antes de gerar a saída;
- Outras tarefas relacionadas:
 - Sumarização de textos;
 - Simplificação de textos;
 - Geração de Diálogo etc.
- Porta de entrada para os *Transformers*



Universidade de Brasília

Departamento de Ciência da Computação



Arquiteturas Encoder-Decoder

Curso ENAP - Processamento de Linguagem Natural

Prof. Dr. Vinícius Ruela Pereira Borges

`viniciusrpb@unb.br`

Brasília-DF, 2024