PROCESSAMENTO DE LINGUAGEM NATURAL

Similaridade Semântica

TÓPICOS

1. Introdução
2. Semântica
3. Similaridade Semântica
4. Estado da arte

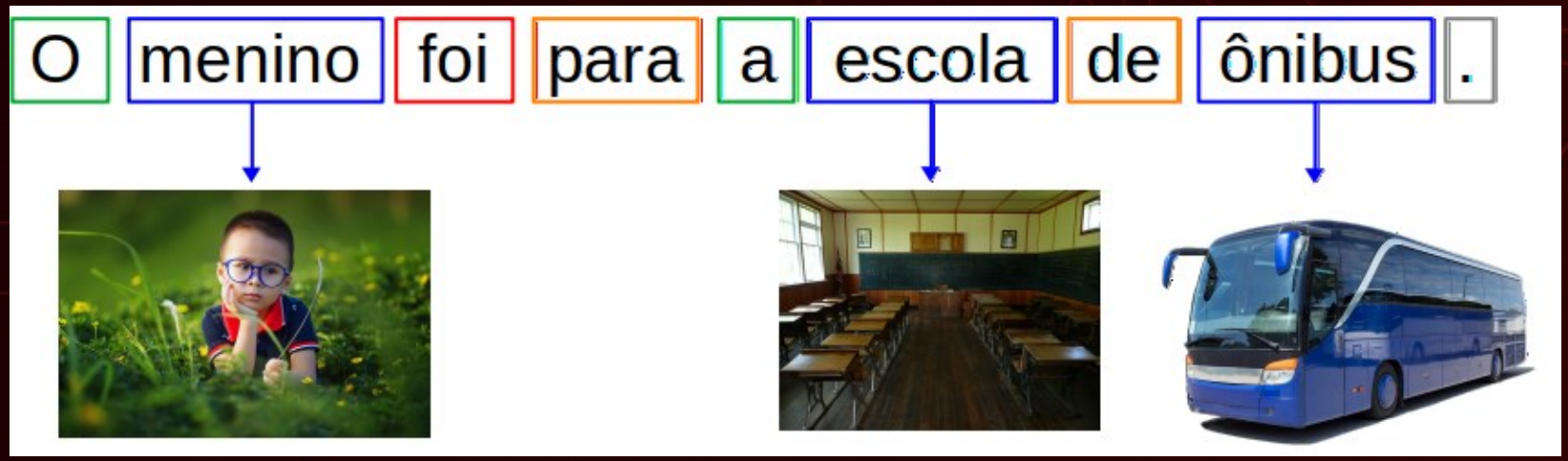
LINGUAGENS

- O que é a linguagem?

“Sistema de símbolos de um vocabulário que, quando colocados numa determinada ordem e expressos num determinado contexto, emitem um significado.”

PLN

- **Quão difícil é processar automaticamente a linguagem natural?**



Análise Semântica
Semântica Lexical

NÍVEIS LINGUÍSTICOS

- **Semântica**
  - Estudo dos significados
  - **Semântica lexical**
    - Entendimento do significado das unidades linguísticas (ex.: **escola** e **ônibus**)
  - **Semântica composicional**
    - Entendimento do significado de unidades que se agrupam em uma frase (ex.: **escola de inglês**)

SEMÂNTICA LEXICAL

- **Polissemia**
  - Quando a mesma palavra tem significados relacionados
  - Ex.: “letra”
- **Homonímia**
  - Quando a mesma palavra tem significados não relacionados
  - Ex.: “manga”

SEMÂNTICA LEXICAL

- **Relações**
  - Sinonímia
    - cômico ~ engraçado
    - palavra ~ vocábulo
  - Antonímia
    - bom ~ ruim
    - amar ~ odiar

SEMÂNTICA LEXICAL

- **Relações**
  - Hiperonímia / Hiponímia
    - fruta → maçã
    - veículo → carro
  - Holonímia / Meronímia
    - carro // roda
    - cadeira // pé

é-um is-a

parte-de part-of

SEMÂNTICA COMPOSICIONAL

- O que é?
  - O significado de uma sentença depende dos itens lexicais que a compõem
  - O significado de uma MWE composicional depende dos itens lexicais que a compõem
- Princípio de Composicionalidade
  - O significado de um constituinte sintático é derivado exclusivamente do significado de seus constituintes imediatos

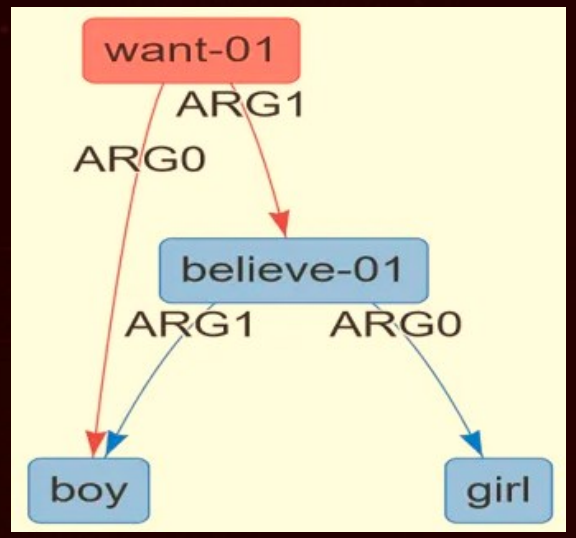
SEMÂNTICA COMPOSICIONAL

- Formalismos de representação
  - Lógica de Primeira Ordem
    - Predicados + Variáveis + Quantificadores + Conectivos lógicos determinam a semântica
    - Ex.: O menino foi para a escola de ônibus.
    - ir(menino, escola) ^ modo(ir, ônibus)

SEMÂNTICA COMPOSICIONAL

- **Formalismos de representação**
  - ***Abstract Meaning Representation***

The boy wants the girl to believe him.
The boy wants to be believed by the girl.



Fonte: https://medium.com/@sroukos/semantic-parsing-using-abstract-meaning-representation-95242518a380

SIMILARIDADE TEXTUAL

- Similaridade: verificar o quão “próximos” são dois fragmentos de texto a partir do (1) significado e de sua (2) estrutura
- (1) similaridade semântica
- (2) similaridade léxica

- Medidas vistas na semana 2
  - Similaridade textual / léxica

SIMILARIDADE SEMÂNTICA

Frase 1 — "O rato come o inseto"
Frase 2 — "O inseto come a comida do rato"

Frase 1 — ["O", "rato", "come", "o", "inseto"]
Frase 2 — ["O", "inseto", "come", "a", "comida", "do", "rato"]

- **Abordagem baseada em ontologias**
- **Abordagem baseada no índice de informações compartilhadas**
- **Abordagem baseada em características**
- **Abordagem híbrida (algum tipo de combinação das três anteriores)**

BASEADA EM ONTOLOGIAS

- Ontologia: é um sistema de descrição abstrata que entende a constituição de conhecimento de certo domínio pela organização de conceitos de maneira hierárquica, descrevendo os relacionamentos entre os conceitos usando um número pequeno de descritores relacionais e vocabulário padronizado para representar as entidades do domínio.

- A similaridade semântica entre palavras é medida com base em recursos semânticos explorando o conhecimento existente dentro desses recursos.

WORDNET

- A WordNet é o recurso de ontologia mais popular e amplamente utilizado na medição de similaridade baseada em conhecimento.
- Grande banco de dados léxicos de um projeto de pesquisa desenvolvido pela Univ. de Princeton que organiza substantivos, verbos, advérbios e adjetivos em um conceito de relações semânticas, chamado de conjuntos de sinônimos.

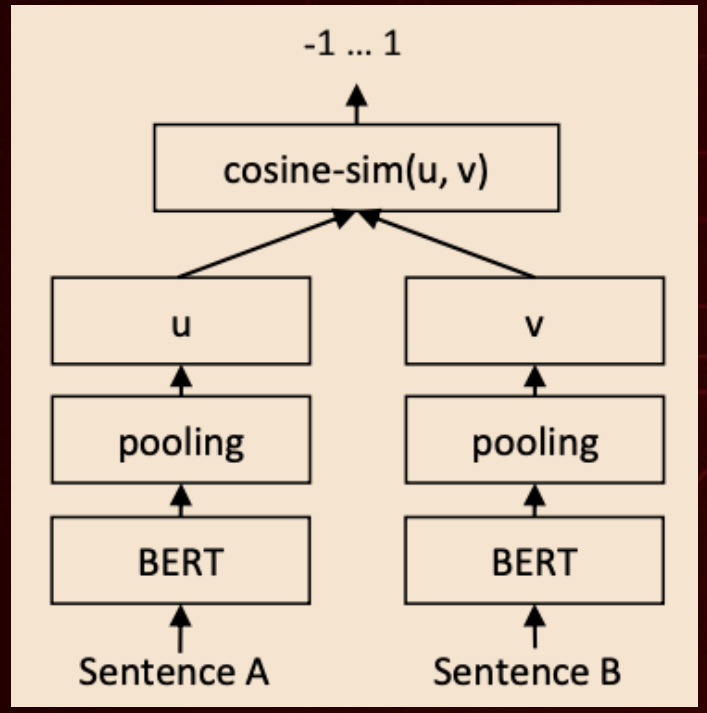
Wordnet	Itens lexicais				
	Substantivo	Verbo	Adjetivo	Advérbio	Total
WN.PT 1.0	9.813	633	485	0	10.931
MWN.PT v1	16.000	0	0	0	16.000
WN.BR	17.000	10.910	15.000	1.000	43.910
Onto.PT 0.6	97.531	32.958	34.392	3.995	168.876
OpenWN-PT	43.996	3.914	5.422	1.388	54.720
UfesWN.BR 1.0	20.646	3.769	9.066	1.498	34.979
PULO	10.260	4.032	3.166	173	17.631
WN.Pr 3.0	119.034	11.531	21.538	4.481	156.584

SIMILARIDADE SEMÂNTICA

- Ontologias
  - Baseadas em arestas: Pekar et al., Cheng and Cline, Wu et al. ...
  - Baseadas em nó: Resnik, Lin, Maguitman, Menczer, Roinestad and Vespignani, Jiang and Conrath, Align, Disambiguate, and Walk
  - Pairwise
  - Groupwise
- Estatísticas: **LSA, PMI, NGD, SSA, SimRank...**
- ***Semantics-based similarity***
- ***Semantics Similarity Networks***
- https://en.wikipedia.org/wiki/Semantic_similarity

ESTADO DA ARTE

- *Transformers* para codificar sentenças e obter seus *embeddings* e, em seguida, usar uma métrica de similaridade (por exemplo, similaridade de cosseno) para calcular sua pontuação de similaridade.



- SBERT – *Sentence-Transformers*
  https://www.sbert.net/docs/usage/semantic_textual_similarity.html

CALCULANDO SIMILARIDADE

Colab - SBERT

```
# instalando dependências
!pip install transformers
!pip install sentence-transformers

# importando pacotes
from sentence_transformers import SentenceTransformer, util
import numpy as np

# selecionando e inicializando o modelo
model = SentenceTransformer('stsb-roberta-large')
```

CALCULANDO SIMILARIDADE

Colab - SBERT

```
sentence1 = "I gosto de Python porque possso construir aplicações de IA"
sentence2 = "I gosto de Python porque posso analisar de dados"

# encode sentences to get their embeddings
embedding1 = model.encode(sentence1, convert_to_tensor=True)
embedding2 = model.encode(sentence2, convert_to_tensor=True)

# compute similarity scores of two embeddings
cosine_scores = util.pytorch_cos_sim(embedding1, embedding2)

print("Sentença 1:", sentence1)
print("Sentença 2:", sentence2)
print("Score de similaridade:", cosine_scores.item())
```

CALCULANDO SIMILARIDADE

Colab - SBERT

```
sentences1 = ["I gosto de Python porque possso construir aplicações de IA",
              "O gato senta no chão"]
sentences2 = ["I gosto de Python porque posso analisar de dados",
              "O gato caminha na calçada"]

# encode list of sentences to get their embeddings
embedding1 = model.encode(sentences1, convert_to_tensor=True)
embedding2 = model.encode(sentences2, convert_to_tensor=True)

# compute similarity scores of two embeddings
cosine_scores = util.pytorch_cos_sim(embedding1, embedding2)

for i in range(len(sentences1)):
    for j in range(len(sentences2)):
        print("Sentença 1:", sentences1[i])
        print("Sentença 2:", sentences2[j])
        print("Score de similaridade:", cosine_scores[i][j].item())
        print()
```

O QUE VIMOS?

- Introdução
- Semântica
- Similaridade Semântica
- Estado da arte

PRÓXIMA VIDEOAULA

- **Análise de Sentimentos**

REFERÊNCIAS

- Curso de Processamento de Linguagem Natural
  - Profa. Helena Caseli (UFSCar)

- Curso de Processamento de Linguagem Natural
  - Prof. Thiago Pardo (ICMC-USP)