EXPLORING TRANSFORMERS FOR RANKING PORTUGUESE SEMANTIC RELATIONS

Hugo Gonçalo Oliveira, hroliv@dei.uc.pt
CISUC, DEI, University of Coimbra, Portugal

Introduction

- Knowledge Bases
  - Well-defined relations, represented by triples (arg1, related-to, arg2), interpretable
  - Manual creation → time-consuming, coverage issues
  - Automatic creation → noisy
  - Approaches for computing the confidence of distinctions [1, 2]
- Transformer Language Models (TLMs)
  - e.g., GPT, BERT
  - Unprecedented advances in NLP
  - Encode much linguistic and world knowledge
- Language models as Knowledge Bases [3]
  - Generate text after prompts
  - Predict masked tokens
  - Compute the likelihood of sequences of text
  - Which may express semantic relations?
  - e.g., A dog is a mammal; A knife is used for cutting

Experimentation

- TLMs for ranking lexico-semantic relations, in Portuguese
  - Filter out very specific / incorrect relation instances
  - Instances from ten lexical resources [4]
    - 892,693 instances, some extracted automatically
    - 16 types: synonymy (4), hypernymy (2), part (2), member, purpose (2), causation, place, property (2)
    - Variable utility
    - prototypical / widely accepted (tree hypernym of oak, to_cook purpose of oven)
    - very specific (cdstore place-of-elvis-presley-ctf, give_to girlfriend purpose of kitty)
    - underspecified / incomplete / incorrect arguments (possessive said-about to make, various causes contest)
  - Weighted by the number of resources they are in (Res = 1–10)
  - Exploit two available TLMs
    - BERTImbus [5] (base), BERT pre-trained for Portuguese
    - GPTtuguese-2*, GPT2-small fine-tuned for Portuguese
  - Approach:
    1. handcraft textual templates that transmit / are compatible with relations of the target types (see Table below)
    2. instantiate the templates for each instance, according to its type
    3. use a TLM for computing the likelihood of the resulting sequences.
- More prototypical / reliable instances have higher scores?

<table>
<thead>
<tr>
<th>Relation</th>
<th>Pattern</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dizer</td>
<td>a dizer para dizer</td>
<td>1.00</td>
</tr>
<tr>
<td>dizer</td>
<td>a dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>1.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
<tr>
<td>dizer</td>
<td>dizer para dizer</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- weight = 0.00 normalised to the 1-10 interval
- For each model, BERT (B) and GPT (G), and pattern (1-3)
- Plus maximum (Mx) and average (Av) weight

Analysis

- Weights are sensitive to: relation + frequency of words
  - Sequences with words that the TLM has seen more times get higher weights, even if the sequence is not semantically-coherent.
  - W(coachman ou outro animal) → W(coachman ou outro animal) → W(esquilo ou outro animal)

Answering Similarity Tests

- node2vec [6] embeddings of six networks
  - No weights, with Res, Mx(B), Av(B), Mx(G), Av(G)
  - dimen.sions = 64, walk.len = 3, walk.lengths = [10,100,200], walk.lengths = [10,30]
  - Used to answer Portuguese adaptations of similarity tests
  - SimLex-999, WordSim-353 [7], PT-65 [8]
  - No noticeable differences...

Wrap up

- TLMs can be used for filtering out noisier examples, e.g., in the automatic creation of knowledge bases
  - Trend is that weights are lower for instances with long and specific arguments
  - No impact when computing semantic similarity
- Better-suited tasks?
  - Weighted PT-LKB instances available

Acknowledgements

This work was financially supported by national funds through FCT, within the scope of project CSIC-ID/CEC/03326/2020 and by FEDER, through the Regional Operational Program Centre 2020. It is also developed in the scope of COST Action CA18292 Nexus Linguarium, supported by COST (European Cooperation in Science and Technology). http://www.cisuc.uc.pt

References