Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2573–2582
Marseille, 20-25 June 2022

Exploring Transformers for Ranking Portuguese Semantic Relations

Hugo Gonçalo Oliveira
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal,
hroliv@dei.uc.pt

Abstract

We explored transformer-based language models for ranking instances of Portuguese lexico-semantic relations. Weights were based on the likelihood of natural language sequences that transmitted the relation instances, and expectations were that they would be useful for filtering out noisier instances. However, after analysing the weights, no strong conclusions were taken. They are not correlated with redundancy, but are lower for instances with longer and more specific arguments, which may nevertheless be a consequence of their sensitivity to the frequency of such arguments. They did also not reveal to be useful when computing word similarity with network embeddings. Despite the negative results, we see the reported experiments and insights as another contribution for better understanding transformer language models like BERT and GPT, and we make the weighted instances publicly available for further research.

Keywords: semantic relations, lexical patterns, transformer models, BERT, GPT

1. Introduction

Even though distributional semantics and deep learning are the current trend in Natural Language Processing (NLP), research on the automatic acquisition of semantic relations from large corpora and semi-structured sources has a long history, which, among others, lead to the development of several systems for Open Information Extraction from the Web (Etzioni et al., 2008), as well as large knowledge bases like DBPedia (Auer et al., 2007), YAGO (Tanon et al., 2020), or BabelNet (Navigli and Ponzetto, 2012). In opposition to distributional and neural language models, where words and sequences are represented by vectors of numbers, in the previous, relations are represented by triples of the kind 'arg_1 related-to arg_2', and are thus interpretable by humans. However, the automatic acquisition of relations can be a noisy process, and it is not always straightforward to discriminate between good extractions and those that are irrelevant or simply incorrect. To help with the latter, there has been work on computing the confidence of extractions, e.g., based on the occurrence of the relation arguments in a large collection of text (Cederberg and Widdows, 2003; Downey et al., 2005; Cimiano and Wenderoth, 2007).

On the other hand, the adoption of models based on transformers (hereafter, TLMs), like GPT (Brown et al., 2020) and BERT (Devlin et al., 2019), lead to unprecedented advances in a broad range of NLP tasks. Since the latter encode much linguistic and world knowledge, some authors (Petroni et al., 2019; Haviv et al., 2021) show that, to some extent, they can be used as knowledge bases, e.g., when used to fill blanks in given text (see a recent review on the topic (AlKhamissi et al., 2022)). Furthermore, as it happens for traditional language models, TLMs can be used for computing the likelihood of given sequences of text. In principle, if given sequences express the target relation instances in natural language, this process could be seen as a shortcut for computing the confidence of such instances.

This is what we explore in this paper, though focusing on the Portuguese language and in lexico-semantic relations, which are those one would expect to find in a dictionary or in a resource like WordNet (Fellbaum, 1998). The main contribution of this work is thus in the scope of the automatic creation of lexical knowledge bases. Our starting point is a set of relation instances obtained from ten lexical resources (PT-LKB), and two TLMs for Portuguese, one based on BERT and another on GPT. Since the instances have variable quality and utility, we aim at exploring the TLMs for ranking those instances according to their prototypicality. This would be useful for filtering out less useful (e.g., very specific) or incorrect relations. Inspired by earlier work on relation extraction, we construct sequences that transmit each instance and use the TLMs for computing their likelihood, based on the loss of the model. After this, we analyse the resulting weights, including their relationship to the number of resources each instance was obtained from, and by inspecting the top and bottom-ranked instances. Finally, we use PT-LKB with weights by different TLMs for answering similarity tests automatically. Our conclusions so far are that the new weights provide a new distinct dimension, which can be used to filter out some very specific relations. At the same time, we noted that they are very sensitive to the frequency of the relation arguments and do not lead to improvements in the computation of semantic similarity. Yet, in addition to the previous results, we see the reported insights as another contribution for better understanding TLMs and what we can do with them.

In the remainder of the paper, we overview related work on relation extraction and ranking; we describe the experimentation setup, focused on the weighting process; we give and discuss some insights on the results; and, before concluding, we report the performance of the weighted networks in similarity tests.

2. Related Work

Interest in the automatic acquisition of semantic relations from text has grown especially since the transition to the so-called Web 2.0, which enabled virtually anyone to publish content, resulting in large quantities of text easily accessible. Much related work is inspired by Hearst (1992), where a set of lexico-syntactic patterns was proposed for extracting hyponymy instances that could be used for enriching knowledge bases like WordNet (Fellbaum, 1998). Yet, to minimise human intervention and increase the quantity of extracted relations, automatic procedures were proposed, e.g., based on distant (Snow et al., 2005) or weak supervision (Pantel and Pennacchiotti, 2006), either focused on a closed set of relation types, or following the paradigm of Open Information Extraction (OIE) (Etzioni et al., 2008), where virtually every possible relation is extracted. By broadening the set of considered patterns, which can be learned automatically, such approaches lead to more but also noisier extractions. Therefore, some works focused on scoring extractions according to their reliability, which enabled to increase precision by filtering out some unreliable extractions. A straightforward approach is based on the semantic similarity of the relation arguments (Cederberg and Widdows, 2003) or other co-occurrence measures (Cimiano and Wenderoth, 2007), computed from corpora or using a Web search engine. This however does not consider the relation itself. For that, the actual patterns where the arguments occur have to be considered (Pantel and Pennacchiotti, 2006; Costa et al., 2011). And here, besides the number of times the arguments were found with one of the target patterns, redundancy has shown to be an important cue, i.e., instances extracted from different sources or using different patterns should be more reliable. Considering the previous, a probabilistic model was developed (Downey et al., 2005), and measures were proposed for combining simple co-occurrence with the occurrence in target patterns (Bollegala et al., 2007).

As it happened for other NLP tasks, the state-of-the-art on relation extraction from text currently relies on deep learning, where the task is either framed as a sequence labelling — e.g., based on bidirectional LSTM networks (Stanovsky et al., 2018); or on transformers like BERT (Ro et al., 2020) — or a generation problem — e.g., an LSTM encoder-decoder that generates relation instances from given sentences (Cui et al., 2018), with training examples bootstrapped from a more traditional OIE system (Mausam, 2016); or a BART model, pre-trained in sentences from Wikipedia abstracts and entailed Wikidata relations (Cabot and Navigli, 2021).

The previous are all supervised approaches, trained specifically for relation extraction. An alternative is to acquire relation instances from pre-trained language models, including static word embeddings — e.g., unsupervisedly (Chang et al., 2018), or supervisedly, based on a set of analogies (Drozd et al., 2016)) — or TLMs — e.g., starting with a small number of patterns and seeds (Bouraoui et al., 2020), or based on predefined lexical patterns (Petroni et al., 2019), to some extent similar to those used for relation extraction from corpora.

Specifically for Portuguese, there are several OIE systems, most of which based on rules that consider the part-of-speech tags and dependency parsing (Glauber et al., 2019) or chunking (Sena and Claro, 2020), and, more recently, neural approaches (Cabral et al., 2022). There is also recent work on the acquisition of lexico-semantic relations from static word embeddings (Gonçalo Oliveira et al., 2020), hyponyms (Paes, 2021) and other relations from BERT (Gonçalo Oliveira, 2022). Also for Portuguese, instances of lexico-semantic relations have been acquired from several lexical resources, and weighted according to redundancy, i.e., the intuition is that more reliable and useful an instance is, the more resources it is in (Gonçalo Oliveira, 2018).

3. Experimentation

We aim at exploiting Portuguese TLMs for ranking instances of Portuguese lexico-semantic relations acquired from ten lexical resources, hereafter PT-LKB (Gonçalo Oliveira, 2018). The previous resources include wordnets and thesauri, some of which created (semi-)automatically, and dictionaries, where relation instances were extracted automatically from. Therefore, those instances have variable utility and contain a portion of incorrect extractions as well, i.e., they go from widely accepted prototypical instances (e.g., *tree* hypernym-of *oak*[1]; *to_cook* purpose-of *oven*) to others very specific (e.g., *cd_store* place-of *elvis_presley_cd*; *give_to_girlfriend* purpose-of *kitty*) or with underspecified / incomplete (e.g., *possessive* said-about *to_make*) or incorrect (e.g., *various* causes *contest*) arguments. They are currently weighted according to the number of resources they were obtained from (hereafter, Res weight). This may help to filter out some undesirable instances, but they are limited to discrete values in the 1-10 interval and to the contents of the resources, which may not reflect how language is actually used. An alternative would be to adopt some of the approaches enumerated in Section 2, where instances are scored according to occurrences of their arguments on the Web. Yet, we see TLMs as a potential shortcut for the previous: they are trained in large quantities of text and encode knowledge about language and its usage.

For this purpose, we exploit two TLMs, namely BERT (Devlin et al., 2019) and GPT (Radford et al.,

[1]Relation instances in PT-LKB are in Portuguese but, for the sake of simplicity, we use rough translations in these examples.

2019), respectively pre-trained and fine-tuned for Portuguese. BERT is pre-trained as a masked language model (i.e., it can predict masked tokens based on the left and right context) and GPT can be used as a traditional language model (i.e., it generates text following given sequences). The adopted approach follows three main steps:

1. handcraft a set of textual templates based on patterns that transmit relations of a target type;
2. instantiate the templates for each instance of the corresponding type;
3. use a TLM for computing the likelihood of the sequence resulting from the previous step.

Our hypothesis is that, if the given sequence instantiates a semantic relation, the computed score will be inversely proportional to the perplexity, a hint of the likelihood of the sequence, and thus of the instantiated relation. In other words, higher scores should be given to more frequent relations, potentially more prototypical, and also more reliable.

This section presents the models used in our experimentation and how they were used for scoring the relations. This is followed by an overview of the target relation types, including examples of instances, and an enumeration of the considered lexical patterns.

3.1. Models

Two TLMs were used in this work, both available through the Hugging Face `transformers` library[2]: BERTimbau (Souza et al., 2020) (base), a BERT model pretrained for Portuguese; and GPorTuguese-2, a GPT2 for Portuguese[3], more precisely, GPT2-small fine-tuned with 1GB of text in Portuguese.

These models are significantly different, but they are both based on the transformer architecture. So, after loading them, the likelihood of a sequence of text can be approximated by the exponential of the loss of the model for its tokens, which is what we do. For BERT, however, it is advisable that the special tokens `[CLS]` and `[SEP]` are added respectively to the beginning and to the end of the sequence[4]. A similar approach has recently been adopted for answering Portuguese multiple-choice cloze tests automatically (Gonçalo Oliveira, 2021).

3.2. Relations

PT-LKB contains 938,846 instances, represented by `arg-1 related-to arg-2`, where `arg-1` and `arg-2` are Portuguese words and `related-to` is the name of a relation type. Several types of lexico-semantic relations are covered and their distribution is highly imbalanced. In this work, we focus only on the 16 types for which there are at least 5,000 instances, which accounts for 862,693 instances, about 92% of all the instances. These are illustrated in Table 1, where we include the original name of the relation, in Portuguese, followed by an English translation including the part-of-speech (PoS) of the expected arguments, the number of available instances, and an example, also in Portuguese (original) and English. Note that the names of the relation types consider not only the meaning of the relation, but also the PoS of its arguments. This means that, for instance, there are four types of synonymy, respectively between nouns (n), verbs (v), adjectives (adj) and adverbs (adv).

3.3. Lexical Patterns

For each of the considered relation types, we handcrafted a set of templates based on lexical patterns that transmit these relations, illustrated in Table 2, where A_1 and A_2 are to be replaced by the first and the second argument of the instance, respectively. This means that, for each instance (e.g., *árvore* HIPERONIMO_DE *carvalho*), weights are computed from the loss of the models, given each sequence obtained with the templates for the relation type (e.g., '*árvore é hiperónimo de carvalho*', '*carvalho ou outra árvore*', '*carvalho é um tipo de árvore*').

For some relations (synonymy, hypernymy, part-of, purpose-of), we selected the top-performing patterns in previous work (Gonçalo Oliveira, 2022), where BERTimbau was used for discovering instances by predicting masked tokens. When the top-3 contained a group of very similar patterns, we only used the first, skipped the others, and used the following patterns. This happened, for instance, for patterns that only differed in the gender of a determiner, and means that, as in previous work (Paes, 2021), determiners will not be inflected according to the gender of the arguments (e.g., '*outro*' for HIPERONIMO_DE or '*uma*' for FINALIDADE_DE). Empirically, we also noted that this only had a minor impact in the (relative) computed scores, while enabling to broaden the variability in the used patterns. For the remaining relations, we tried to consider patterns that would typically be used to transmit each relation. Yet, as in some cases it would be virtually impossible to select patterns that always transmit the relation, we also used patterns that were simply compatible with the relation. We did this while trying to use different enough patterns, i.e., covering different constructions.

Once the loss is computed for all resulting sequences, each relation instance will have six new weights, i.e., three for each pattern times the two models (BERT and GPT). Yet, as the patterns used for different relations are significantly different, the computed scores are not comparable. Therefore, we decided to compute the logarithm of these scores and normalise them to the 0–10 interval, the same of Res, but with continuous values.

[2] `https://huggingface.co/transformers/`

[3] `https://huggingface.co/pierreguillou/gpt2-small-portuguese`

[4] After experimenting with and without `[CLS]` and `[SEP]`, we decided to always add them.

Relation Name		Instances	Example
SINONIMO_N_DE	(n synonym-of n)	155,224	*pedinte – mendigo* (beggar, mendicant)
SINONIMO_V_DE	(v synonym-of v)	127,779	*agarrar – pegar* (grab, catch)
SINONIMO_ADJ_DE	(adj synonym-of adj)	92,028	*vulgar – ordinário* (vulgar, ordinary)
SINONIMO_ADV_DE	(adv synonym-of adv)	6,583	*porventura – talvez* (perhaps, possibly)
HIPERONIMO_DE	(n hypernym-of n)	204,860	*árvore – carvalho* (tree, oak)
HIPERONIMO_ACCAO_DE	(v hypernym-of v)	108,991	*alterar – modificar* (change, modify)
PARTE_DE	(n part-of n)	19,109	*degrau – escada* (step, stairs)
PARTE_DE_ALGO_COM_PROP	(n part-of adj)	5,675	*força – robusto* (strength, robust)
MEMBRO_DE	(n member-of n)	12,628	*carta – baralho* (card, deck)
FINALIDADE_DE	(v purpose-of n)	23,697	*fumar – charuto* (smoke, cigar)
FAZ_SE_COM	(n purpose-of n)	8,547	*condução – cano* (conduction, pipe)
ACCAO_QUE_CAUSA	(v causes n)	12,137	*poupar – poupança* (save, savings)
LOCAL_ORIGEM_DE	(n place-of n)	18,454	*equador – equatoriano* (ecuador, ecuadorian)
DIZ_SE_DO_QUE	(adj said-of-what v)	28,390	*dependente – depender* (dependable, depend)
DIZ_SE_SOBRE	(adj said-about n)	22,385	*mítico – mito* (mythical, myth)
PROPRIEDADE_SEMELHANTE_A	(adj similar-to adj)	16,206	*adjacente – próximo* (adjacent, close)

Table 1: Lexical patterns indicating lexico-semantic relations.

Relation	Pattern	
SINONIMO_N_DE	A_1 é o mesmo que A_2	(A_1 is the same as A_2)
	A_1 é sinónimo de A_2	(A_1 is a synonym of A_2)
	A_1 é igual a A_2	(A_1 is equals to A_2)
SINONIMO_V_DE	A_1 é o mesmo que A_2	(A_1 is the same as A_2)
	A_1 é a mesma coisa que A_2	(A_1 is the same thing as A_2)
	A_1 é sinónimo de A_2	(A_1 is a synonym of A_2)
SINONIMO_ADJ_DE	A_1 é o mesmo que ser A_2	(A_1 is the same as being A_2)
	A_1 é o mesmo que A_2	(A_1 is the same as A_2)
	A_1 é sinónimo de A_2	(A_1 is a synonym of A_2)
SINONIMO_ADV_DE	A_1 é o mesmo que A_2	(A_1 is the same as A_2)
	A_1 é sinónimo de A_2	(A_1 is a synonym of A_2)
	fazer A_1 é o mesmo que fazer A_2	(to do A_1 is the same as to do A_2)
HIPERONIMO_DE	A_1 é hiperónimo de A_2	(A_1 is a hypernym of A_2)
	A_2 ou outro A_1	(A_2 or other A_1)
	A_2 é um tipo de A_1	(A_2 is a type of A_1)
HIPERONIMO_ACCAO_DE	A_2 e outros modos de A_1	(A_2 and other modes of A_1)
	A_2 e outras maneiras de A_1	(A_2 and other manners of A_1)
	A_2 ou outras maneiras de A_1	(A_2 or other manners of A_1)
PARTE_DE	A_2 tem A_1	(A_2 has A_1)
	A_2 possui A_1	(A_2 possesses A_1)
	A_1 do A_2	(A_1's A_2)
PARTE_DE_ALGO_COM_PROP	A_2 porque tem A_1	(A_2 because it has A_1)
	A_2 tem A_1	(A_2 has A_1)
	A_1 do que é A_2	(A_1 of what is A_2)
MEMBRO_DE	A_1 é membro de A_2	(A_1 is a member of A_2)
	A_1 pertence a A_2	(A_1 belongs to A_2)
	A_1 faz parte de A_2	(A_1 is part of A_2)
FINALIDADE_DE	preciso de uma A_2 para A_1	(I need a A_2 for A_1)
	A_2 serve para A_1	(A_1 is for A_2)
	A_2 é usado para A_1	(A_1 is used for A_2)
FAZ_SE_COM	A_1 faz-se com A_2	(A_1 is made with A_2)
	A_2 para fazer A_1	(A_2 to make A_1)
	A_2 para A_1	(A_2 for A_1)
ACCAO_QUE_CAUSA	A_1 causa A_2	(A_1 causes A_2)
	A_2 resulta de A_1	(A_2 results from A_1)
	A_2 é um efeito de A_1	(A_2 is an effect of A_1)
LOCAL_ORIGEM_DE	A_2 vem de A_1	(A_2 comes from A_1)
	A_2 é de A_1	(A_2 is from A_1)
	A_2 de A_1	(A_2 of A_2)
DIZ_SE_DO_QUE	A_1 diz-se do que A_2	(A_1 is said of what A_2)
	A_1 diz-se daquele que A_2	(A_1 is said of the one that A_2)
	A_1 porque A_2	(A_1 because A_2)
DIZ_SE_SOBRE	A_1 diz-se sobre A_2	(A_1 is said about A_2)
	A_1 é relativo a A_2	(A_1 is relative to A_2)
	A_2 devido a A_1	(A_2 because A_2)
PROPRIEDADE_SEMELHANTE_A	A_1 é semelhante a A_2	(A_1 is similar to A_2)
	A_1 é parecido com A_2	(A_1 looks like A_2)
	A_1 parece A_2	(A_1 seems like A_2)

Table 2: Lexical patterns indicating lexico-semantic relations.

Furthermore, we combine the scores of the patterns for each model, which results in four additional weights: the maximum of the three weights and their average, for each model. These final changes are illustrated in Tables 3 and 4, where example instances are respectively shown with the originally computed weights (exponential of the loss) and after normalisation plus computation of the combined weights. Res stands for the number of resources the instance was obtained from, Bn stands for the nth pattern for BERT, Gn for the nth pattern for GPT. In Table 4, Mx(x) stands for the maximum of the x patterns and Av(x) for the average of the x patterns.

Instance	Res	$B1$	$B2$	$B3$	$G1$	$G2$	$G3$
feito SINONIMO_ADJ_DE *concluído*	3	26.19	49.29	73.02	138.16	165.2	413.67
cozer FINALIDADE_DE *panela*	2	16.43	44.17	23.45	160.67	476.8	158.56
sonar PARTE_DE *submarino*	1	48.39	70.93	49.13	4802.38	706.29	14161.6

Table 3: Example instances and their weights as originally computed by the models.

Instance	Res	$B1$	$B2$	$B3$	$G1$	$G2$	$G3$	Mx(B)	Av(B)	Mx(G)	Av(G)
feito SINONIMO_ADJ_DE *concluído*	3	5.28	5.56	6.48	5.37	5.59	6.34	6.12	5.86	6.34	5.99
cozer FINALIDADE_DE *panela*	2	4.11	4.75	4.62	4.54	5.18	4.74	4.75	4.64	5.18	4.93
sonar PARTE_DE *submarino*	1	4.69	4.50	4.56	5.68	4.25	5.84	4.54	4.95	5.63	5.57

Table 4: Example instances and their weights after normalisation and combination.

4. Insights

After weighting all the instances, we try to get some conclusions on the utility of this process. First, we analyse how comparable the new weights were to the previous Res weight, based on the number of resources. Then, we check to what extent the instances with the higher weights are actually better (i.e., more prototypical) than those with lower weights.

4.1. Weight Correlation

To better understand the relation between different weights, the Pearson correlation was computed for all pairs of weights. The higher the correlation, the stronger the linear relationship between the weights. For instance, a high correlation between Res and any other weight would mean that the transformers could be used to simulate redundancy / the presence in different resources, and were thus an alternative to the previous weight, e.g., in the typical case when there are not many lexical resources available. A lower correlation would mean that they are measuring something completely different, either reflecting the gap between relations in lexical resources versus relations actually used in language, or just because the weights do not have the expected meaning. Moreover, given the similarity of the approach and of the models, it is expectable that the weights by BERT are correlated with those by GPT.

Table 5 shows correlations between the different pairs of weights, for all relation types. We focus on the combination weights (Mx and Av) because they consider different patterns and should be more generalisable.

Looking at the overall coefficients, the only meaningful correlation is between the BERT and the GPT weights, as expected, but it is still very weak. All the others are close to zero, meaning no correlation. Considering specific relations, we see that there is a minority of relations contributing more to the weak overall correlation between BERT and GPT weights, especially those between verbs (SINONIMO_V_DE and HIPERONIMO_ACCAO_DE), but there are also several relations with weak to moderate correlation (PARTE_DE, ACCAO_QUE_CAUSA). On the correlation with Res, not much is worth noting, except from weak correlations for five relations (SINONIMO_ADV_DE, HIPERONIMO_ACCAO_DE, PARTE_DE, MEMBRO_DE, PROPRIEDADE_SEMELHANTE_A), both with the maximum and average of the BERT weights. For GPT, no additional correlations were found.

The previous coefficients are highly affected by the low granularity of Res (10 discrete values) when compared to the other weights (continuous from 0 to 10). Moreover, instances with $Res = 1$ make up $\approx 82\%$ of all the instances, which adds noise to the actual correlation. Having this in mind, we also computed the correlation between Res and the average weights for each relation and value of Res, with results in Table 6. Now, there seems to be a strong positive correlation between the BERT-based weights, in opposition to a strongly negative correlation with the GPT weights. Still, looking at specific relations, there are exceptions, starting with the place-of relation, for which BERT weights are negatively correlated to Res.

Figure 1 and Table 7 complement this analysis. The former shows the evolution of the four combination weights for increasing values of Res, while the latter is focused on Av(BERT), and shows the average weights and standard deviation for each relation and value of Res. Even though the correlation between Res and the average BERT weights is now clear, the variation is low, with high standard deviations. This means that there is a significant number of instances for which this correlation does not apply, also contribution to the non-existent correlations overall, reported in Table 5. On the other hand, the average GPT-based is indeed negatively correlated with Res. In opposition to previous work, where this precise model (Gonçalo Oliveira, 2021), using exactly the same method, was the best for selecting the best option for answering cloze questions, these figures suggest that it is not suitable for our purpose. An important difference towards BERTimbau, which might have some impact, is that GPorTuguese-2 results from fine-tuning the original GPT2 with Portuguese text, while BERTimbau was pretrained from scratch for Portuguese.

4.2. Weight Meaningfulness

As previously discussed, having no correlation does not necessarily mean that the new weights are not useful for identifying the best relations, but that they capture a different dimension than Res. For additional insights, we inspected the top and bottom-weighted instances, looking for a trend. In the bottom, we found mostly instances with long multiword expressions as argu-

Relation	Res,Mx(B)	Res,Av(B)	Res,Mx(G)	Res,Av(G)	Mx(B),Mx(G)	Av(B),Av(G)
SINONIMO_N_DE	0.08	0.08	-0.03	-0.04	0.13	0.09
SINONIMO_V_DE	0.07	0.07	-0.12	-0.13	-0.09	-0.13
SINONIMO_ADJ_DE	0.07	0.07	-0.09	-0.09	0.03	0.00
SINONIMO_ADV_DE	0.13	0.13	-0.05	-0.05	0.05	0.06
HIPERONIMO_DE	0.07	0.08	0.02	0.01	0.37	0.39
HIPERONIMO_ACCAO_DE	0.13	0.13	-0.04	-0.05	-0.03	-0.06
PARTE_DE	0.19	0.20	0.08	0.07	0.45	0.44
PARTE_DE_ALGO_COM_PROP	-0.01	-0.01	0.03	0.03	0.36	0.35
MEMBRO_DE	0.14	0.14	0.03	0.01	0.29	0.24
FINALIDADE_DE	0.07	0.07	0.06	0.06	0.32	0.29
FAZ_SE_COM	0.04	0.04	0.07	0.07	0.33	0.32
ACCAO_QUE_CAUSA	0.03	0.02	0.04	0.04	0.45	0.45
LOCAL_ORIGEM_DE	-0.01	0.00	0.02	0.03	0.32	0.27
DIZ_SE_DO_QUE	0.04	0.04	0.05	0.05	0.39	0.36
DIZ_SE_SOBRE	-0.01	-0.02	0.03	0.02	0.19	0.17
PROPRIEDADE_SEMELHANTE_A	0.17	0.17	0.08	0.08	0.32	0.31
All	0.02	0.04	-0.05	-0.04	0.22	0.18

Table 5: Pearson correlation between different weights for different relations.

Relation	Mx(B)	Av(B)	Mx(G)	Av(G)
SINONIMO_N_DE	0.99	0.99	-0.33	-0.64
SINONIMO_V_DE	0.98	0.98	-0.86	-0.92
SINONIMO_ADJ_DE	0.94	0.91	-0.76	-0.65
SINONIMO_ADV_DE	0.82	0.82	-0.23	-0.28
HIPERONIMO_DE	0.99	0.99	-0.69	-0.75
HIPERONIMO_ACCAO_DE	0.98	0.98	-0.90	-0.95
PARTE_DE	0.56	0.64	0.01	0.06
PARTE_DE_ALGO_COM_PROP	0.72	0.70	0.97	0.98
MEMBRO_DE	0.82	0.86	-0.67	-0.82
FINALIDADE_DE	0.97	0.96	0.47	0.16
FAZ_SE_COM	0.88	0.82	0.99	0.98
ACCAO_QUE_CAUSA	0.75	0.73	0.81	0.80
LOCAL_ORIGEM_DE	-0.42	-0.39	0.08	0.01
DIZ_SE_DO_QUE	0.93	0.92	0.94	0.93
DIZ_SE_SOBRE	0.90	0.88	0.37	-0.02
PROPRIEDADE_SEMELHANTE_A	0.94	0.94	1.00	1.00
All	0.95	0.96	-0.88	-0.91

Table 6: Pearson correlation between average weights and Res, for different relations.

	Res								
Relation	1	2	3	4	5	6	7	8	9
SINONIMO_N_DE	3.86±1.0	3.97±0.9	4.05±0.9	4.12±0.9	4.20±0.8	4.26±0.8	4.28±0.8	4.41±0.7	–
SINONIMO_V_DE	4.09±0.9	4.11±0.9	4.24±0.9	4.27±0.9	4.33±0.9	4.36±0.9	4.41±0.9	4.57±0.8	4.63±0.7
SINONIMO_ADJ_DE	3.78±0.8	3.87±0.8	3.92±0.8	3.95±0.8	4.04±0.8	4.15±0.9	4.16±0.8	4.21±0.8	4.12±0.0
SINONIMO_ADV_DE	4.83±1.1	5.03±1.1	5.28±1.1	5.53±1.0	5.77±1.0	5.99±1.2	5.78±1.4	5.54±0.8	–
HIPERONIMO_DE	4.21±1.1	4.45±1.0	4.47±0.9	4.61±0.9	4.76±0.9	4.94±0.7	–	–	–
HIPERONIMO_ACCAO_DE	4.60±1.2	5.21±1.3	5.49±1.2	5.81±1.0	–	–	–	–	–
PARTE_DE	4.18±1.3	5.35±1.3	5.09±1.3	6.49±1.7	5.09±0.0	–	–	–	–
PARTE_DE_ALGO_C..._PROP	4.25±1.2	4.19±1.1	4.44±1.0	–	–	–	–	–	–
MEMBRO_DE	3.95±1.1	4.66±1.1	5.06±0.9	4.82±0.9	–	–	–	–	–
FINALIDADE_DE	3.97±1.0	4.34±1.0	4.53±1.1	4.65±0.8	–	–	–	–	–
FAZ_SE_COM	4.41±1.2	4.61±1.0	4.61±0.9	–	–	–	–	–	–
ACCAO_QUE_CAUSA	3.47±1.1	3.23±0.9	3.52±1.0	4.01±1.1	–	–	–	–	–
LOCAL_ORIGEM_DE	4.32±1.2	3.98±0.9	4.61±1.3	3.66±0.0	–	–	–	–	–
DIZ_SE_DO_QUE	3.63±1.1	3.65±1.0	4.18±1.3	5.09±1.3	–	–	–	–	–
DIZ_SE_SOBRE	4.01±1.0	3.79±0.9	4.14±0.8	4.49±0.8	4.94±0.7	–	–	–	–
PROPRIEDADE_SEM..._A	3.70±1.3	4.52±1.5	4.70±0.8	–	–	–	–	–	–

Table 7: Average weights for different relations and number of resources.

ments[5] (e.g., *ácido_docosa_hexaenóico* SINONIMO_N_DE *ácido_4z_7z_10z_13z_16z_19z_docosa_hexaenoico*, *reino_unido_da_grã-bretanha_e_irlanda_do_norte* MEMBRO_DE *organização_do_tratado_do_atlântico_norte*). This is in line with what we were expecting, i.e., by discarding the bottom-weighted instances, we would also get rid of many instances with so specific arguments that would hardly be of any use.

Since most of the previous had been obtained from a single resource ($Res = 1$), we decided to look at the top and bottom-weighted instances for specific values of Res. For illustrative purposes, Table 10 shows the instances in two resources ($Res = 2$) with top and bottom Av(B) weights. The bottom-weighted still include some instances with multiword expressions, and mostly instances with very specific arguments. On the top-weighted, this judgement is harder to make, but quality is generally better, suggesting that the weights can indeed be useful for filtering out lower quality instances.

4.3. Argument Frequency vs Validity

While performing these experiments, we noted that the computed losses are not only sensitive to the relation

[5] Even though multiword expressions have terms separated by underscores ('_'), before using them for computing weights, underscores were replaced by spaces.

Relation	Top(Av(B))	Bottom(Av(B))
SINONIMO_N_DE	beira–cairel	mesa-de-cabeceira–mesinha-de-cabeceira
SINONIMO_V_DE	cortar–foiçar	empeçonhar–empeçonhentar
SINONIMO_ADJ_DE	instável–lábil	infortunado–mal-afortunado
SINONIMO_ADV_DE	individualmente–particularmente	nem_que_a_vaca_tussa–nem_a_pau
HIPERONIMO_DE	senhora–condessa	búfalo–búfalo_asiático
HIPERONIMO_ACCAO_DE	mover–depor	metamorfosear–acostumar
PARTE_DE	governo–navegação	pára-brisas–aeroplano
PARTE_DE_ALGO_COM_PROP	saúde–válido	pé_de_cor_de_açafrão–crocípede
MEMBRO_DE	esforço–campanha	cebola-albarrã–liliáceas
FINALIDADE_DE	tratar–procurador	tirar_rolha_da_garrafa–saca-rolhas
FAZ_SE_COM	líquido–taleiga	galvanopunctura–agulha
ACCAO_QUE_CAUSA	livrar–livre	empeçonhar–empeçonhamento
LOCAL_ORIGEM_DE	estado–catarinense	freixo-de-espada-à-cinta–freixonita
DIZ_SE_DO_QUE	consecutivo–seguir	pectinibrânquio–ter_brânquia_em_forma_de_pente
DIZ_SE_SOBRE	austríaco–áustria	mnemotécnico–mnemotecnia
PROPRIEDADE_SEMELHANTE_A	essencial–principal	inaceitável–inadmissível

Table 8: Top and bottom-weighted instances in only two resources.

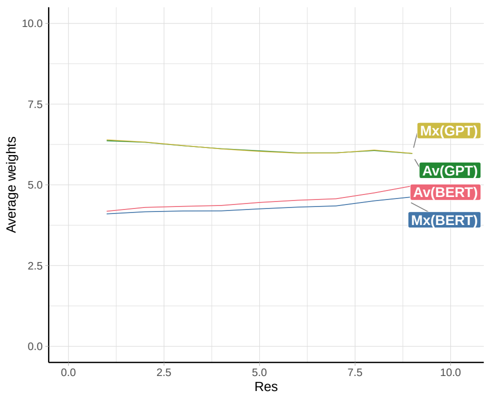

Figure 1: Average weight per number of resources.

Valid	Sequence		e^{loss}
✓	*cachorro ou outro animal*	(dog or other animal)	122.1
✓	*gato ou outro animal*	(cat or other animal)	134.8
✓	*esquilo ou outro animal*	(squirrel or other animal)	53.4
×	*carro ou outro animal*	(car or other animal)	115.1
×	*skate ou outro animal*	(skate or other animal)	40.4
✓	*carro tem rodas*	(car has wheels)	227.8
✓	*moto tem rodas*	(motorcycle has wheels)	198.1
✓	*skate tem rodas*	(skate has wheels)	52.2
×	*gato tem rodas*	(cat has wheels)	270.7
×	*esquilo tem rodas*	(squirrel has wheels)	86.0

Table 9: Weights of sequences in BERTimbau base.

they are transmitting, but also to the words used and their commonality, i.e., sequences that contain words that the TLM has seen more times will get higher weights. This may have a negative impact on the ideal weight attribution, because it makes it hard to discriminate between semantically-valid sentences and well-formed sentences that use frequent words. To illustrate this, we selected two relations with a fixed argument ('x hypernym of *animal*' and '*wheel* part-of x'), and one pattern for each, then instantiated with valid and invalid arguments. For illustrative purposes, we selected arguments with variable frequency in a corpus of Brazilian Portuguese (Berber Sardinha et al., 2009), the variant BERTimbau was pre-trained for, namely: *cachorro* (6,372), *gato* (6,586), *esquilo* (78), *carro* (90,500), *moto* (5,209), *skate* (1,564). Table 9 shows the exponential of the loss in BERTimbau (weights) for the resulting sequences. For the valid arguments, relative weights seem to have some correlation with the frequency of their variable argument (e.g., *cat* and *dog* are more frequent than *squirrel*; *car* is more frequent than *motorcycle* and *skate*). However, even if invalid, when a very frequent word is used as the argument (e.g., *car* or *cat*), the weight is higher than for the less frequent valid argument (*squirrel*) or for all the valid arguments (*car*, *motorcycle*, *skate*).

5. Answering Similarity Tests

In the previous section, we noted some positive insights from weights computed by TLMs, but also issues that made us unsure on their suitability, especially of the GPT-based weights. To get more on the utility of these weights, and on their advantages when compared to no weights or to simply using Res, we used them in a more objective task, for which a benchmark exists.

This task was word similarity and our gold data were adaptations of well-known similarity tests to Portuguese, namely PT-65 (Granada et al., 2014), SimLex-999 and WordSim-353 (Querido et al., 2017), which contain pairs of Portuguese words and their semantic similarity or relatedness score, based on human judgements (e.g., `pássaro grua 0.24` or `menino rapaz 3.58`). The goal was to exploit the network resulting from PT-LKB and the different weights for computing the similarity of every pair in the test, to finally assess the results with the Pearson correlation between the automatically-computed and the gold scores.

A similar approach to that of previous work (Gonçalo Oliveira, 2018) was followed: embeddings were learned from PT-LKB with node2vec (Grover and Leskovec, 2016), while considering the TLM weights, Res, and no weights. Node2vec represents node neighbourhoods in a d-dimensional feature space by applying a biased random walk procedure. We used its implementation available in the node2vec Python library[6] and ran the algorithm with the following parameters: $dimensions = 64$, $walk\_length = 80$,

[6] `https://pypi.org/project/node2vec/`

$num\_walks = 10, window = 3, min\_count = 1$[7]. After this process, each word in the network is represented as a numeric vector of size 64, and the similarity between two words can be given by the cosine of their vectors. So, the goal was to compare the performance of different embeddings in the considered similarity tests. Table 10 presents the Pearson correlations achieved by the embeddings learned for different weights. As it happens to other embedding methods, node2vec is not deterministic, meaning that each run for the same network may result in slightly different vectors. Therefore, for each considered weight, a total of five node2vec models were learned. This enabled us to compute the mean correlation and the standard deviation. We should add that pairs for which one of the arguments was not in the network were ignored (3.5%, 6.3% and 0, respectively in SimLex, WordSim and PT-65).

Coefficients in the table show that no network stood out and no significant improvements could be achieved with any of the weights. Not even with the Res weights when compared to no weights, which is contradictory to results reported in Gonçalo Oliveira (2018), and is possibly caused by minor differences in the experimentation setup (e.g., different implementation of node2vec, different window size, considered relation types and number of runs). So, in this scenario, the weights computed with the TLMs have shown to be no more useful than Res, based on the number of resources. However, results also suggest that the impact of the weight when embedding the network with node2vec is not enough for exposing the differences, and that this experiment was not the best for reaching strong conclusions. Alternative experiments will have to be devised in the future.

6. Conclusion

Inspired by early work on ranking automatically acquired relation instances from text, useful for discarding noisier extractions, we explored recent TLMs for a similar purpose, while avoiding to search directly on large corpora. We focused in Portuguese lexico-semantic relations and weighted the instances in PT-LKB, obtained from ten lexical resources. For each of the 16 relation types considered, three lexical patterns were handcrafted. TLMs for Portuguese were then used for computing the likelihood of the sequences resulting from instantiating the patterns with the relation instances. The latter scores were used as weights. Though not correlated with the number of resources the instances were obtained from, weights were lower for instances with very long and specific arguments, suggesting that weights computed like this can be used for filtering out noisier extractions. This may help in the selection of more prototypical relations from a large set, and be useful for the automatic creation of more reliable knowledge bases.

However, when used for computing semantic similarity, the new weights did not make a difference to using no weights. Towards stronger conclusions, alternative experiments will have to be made in the future. For instance, in the domain of semantic similarity, we may consider pre-discarding low-weighted instances, or try to consider the relation type in the process. Still, we should look for tasks where the impact of the weights is more noticeable. It would also be interesting to test a similar approach in alternative TLMs, not only for Portuguese, but for other languages. For that, in addition to the TLM and instances to weight, a small list of lexical patterns would have to be written for each target relation, which should be straightforward for most relations.

In the meantime, the weighted PT-LKB instances are available from `https://github.com/NLP-CISUC/PT-LexicalSemantics/blob/master/Relations/triplos_pesados_norm.tsv.zip` for anyone willing to use them and possibly complement our conclusions. Despite this negative result, we see the discussed insights as another contribution to better understanding TLMs, what linguistic knowledge they encode, and how we can leverage on it towards better language resources.

[7] Experiments were also made with more walks (e.g., 100, 200) of lower length (e.g., 30), with no clear changes.

Acknowledgements

This work was financially supported by national funds through FCT, within the scope of project CISUC (UID/CEC/00326/2020) and by European Social Fund, through the Regional Operational Program Centro 2020. It is also developed in the scope of COST Action CA18209 Nexus Linguarum, supported by COST (European Cooperation in Science and Technology). `http://www.cost.eu/`.

7. Bibliographical References

AlKhamissi, B., Li, M., Celikyilmaz, A., Diab, M., and Ghazvininejad, M. (2022). A review on language models as knowledge bases. https://arxiv.org/abs/2204.06031.

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 757–766, New York, NY, USA. Association for Computing Machinery.

Bouraoui, Z., Camacho-Collados, J., and Schockaert, S. (2020). Inducing relational knowledge from BERT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7456–7463. AAAI Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A.,

Weights	SimLex-999 N	V	Adj	All	WordSim-353	PT-65
None	0.687±0.004	0.720±0.003	0.810±0.007	0.695±0.003	0.545±0.010	0.867±0.008
Res	0.685±0.004	0.728±0.004	0.829±0.004	0.699±0.002	0.543±0.005	0.881±0.013
Mx(B)	0.678±0.004	0.718±0.005	0.805±0.007	0.689±0.003	0.546±0.011	0.889±0.011
Av(B)	0.678±0.004	0.715±0.007	0.799±0.004	0.686±0.004	0.552±0.013	0.874±0.013
Mx(G)	0.679±0.003	0.723±0.002	0.804±0.004	0.689±0.002	0.545±0.006	0.881±0.013
Av(G)	0.685±0.002	0.721±0.007	0.811±0.003	0.695±0.002	0.544±0.014	0.874±0.009

Table 10: Pearson correlation in similarity tests for embeddings learned with different weights (averages of five node2vec embeddings).

Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cabot, P.-L. H. and Navigli, R. (2021). Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Cabral, B., Souza, M., and Claro, D. B. (2022). PortNOIE: A neural framework for open information extraction for the portuguese language. In *International Conference on Computational Processing of the Portuguese Language*, pages 243–255. Springer.

Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the recall and precision of automatic hyponymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 111–118.

Chang, H.-S., Wang, Z., Vilnis, L., and McCallum, A. (2018). Distributional inclusion vector embedding for unsupervised hypernymy detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 485–495.

Cimiano, P. and Wenderoth, J. (2007). Automatic acquisition of ranked qualia structures from the web. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 888–895.

Costa, H., Gonçalo Oliveira, H., and Gomes, P. (2011). Using the Web to validate lexico-semantic relations. In *Progress in Artificial Intelligence, Proceedings of 15th Portuguese Conference on Artificial Intelligence (EPIA 2011)*, volume 7026 of *LNCS*, pages 597–609, Lisbon, Portugal, October. Springer.

Cui, L., Wei, F., and Zhou, M. (2018). Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. ACL.

Downey, D., Etzioni, O., and Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1034–1041, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Drozd, A., Gladkova, A., and Matsuoka, S. (2016). Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings the 26th International Conference on Computational Linguistics: Technical papers COLING 2016*, COLING 2016, pages 3519–3530.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Glauber, R., Claro, D. B., and de Oliveira, L. S. (2019). Dependency parser on open information extraction for portuguese texts-dptoie and dependentie on iberlef. In *IberLEF@ SEPLN*, pages 442–448.

Gonçalo Oliveira, H., Sousa, T., and Alves, A. (2020). TALES: Test set of Portuguese lexical-semantic relations for assessing word embeddings. In *Procs of ECAI 2020 Workshop on Hybrid Intelligence for Natural Language Processing Tasks (HI4NLP 2020)*, volume 2693 of *CEUR Workshop Proceedings*, pages 41–47. CEUR-WS.org.

Gonçalo Oliveira, H. (2018). Learning word embeddings from Portuguese lexical-semantic knowledge bases. In *Computational Processing of the Portuguese Language - 13th International Conference, PROPOR 2018, Canela, Brazil, September 24-26, 2018, Proceedings*, volume 11122 of *LNCS*, pages 265–271. Springer, September.

Gonçalo Oliveira, H. (2022). Drilling lexico-semantic knowledge in portuguese from BERT. In *Computational Processing of the Portuguese Language – 14th International Conference, PROPOR 2022*, volume 12037 of *LNCS*, pages 387—397. Springer.

Gonçalo Oliveira, H. (2021). Answering fill-in-the-blank questions in Portuguese with Transformer Language Models. In *Proceedings of 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7-9, 2021*, volume 12981 of *LNCS*, pages 739–751. Springer.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Haviv, A., Berant, J., and Globerson, A. (2021). BERTese: Learning to speak to BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3618–3623.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of 14th Conference on Computational Linguistics*, COLING 92, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.

Mausam, M. (2016). Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 4074–4077.

Paes, G. E. (2021). Detecção de hiperônimos com bert e padrões de hearst. Master's thesis, Universidade Federal de Mato Grosso do Sul.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Procs of 21st International Conference on Computational Linguistics and 44th annual meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia. ACL Press.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In *Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473. ACL.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Ro, Y., Lee, Y., and Kang, P. (2020). Multi^{2}OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117, Online, November. Association for Computational Linguistics.

Sena, C. F. L. and Claro, D. B. (2020). PragmaticOIE: a pragmatic open information extraction for portuguese language. *Knowledge and Information Systems*, pages 1–26.

Snow, R., Jurafsky, D., and Ng, A. (2005). Learning syntactic patterns for automatic hypernym discovery. *Advances in neural information processing systems*, 17:1297–1304.

Stanovsky, G., Michael, J., Zettlemoyer, L., and Dagan, I. (2018). Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895.

8. Language Resource References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Berber Sardinha, T., Moreira Filho, J., and Alambert, E. (2009). The brazilian corpus: A one-billion word online resource. In *Mahlberg, M., González-Díaz, V., Smith, C., Proceedings of the Fifth Corpus Linguistics Conference, CL2009, University of Liverpool, UK*, pages 20–23.

Christiane Fellbaum, editor. (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Gonçalo Oliveira, H. (2018). A survey on Portuguese lexical knowledge bases: Contents, comparison and combination. *Information*, 9(2).

Granada, R., Trojahn, C., and Vieira, R. (2014). Comparing semantic relatedness between word pairs in portuguese using wikipedia. In *Computational Processing of the Portuguese Language - 11th International Conference, PROPOR 2014*, LNCS, pages 170–175, Cham. Springer.

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, December.

Querido, A., Carvalho, R., Rodrigues, J., Garcia, M., Silva, J., Correia, C., Rendeiro, N., Pereira, R. V., Campos, M., and Branco, A. (2017). LX-LR4DistSemEval: A collection of language resources for the evaluation of distributional semantic models of Portuguese. *Revista da Associação Portuguesa de Linguística*, (3):265–283.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT models for Brazilian Portuguese. In *Proceedings of Brazilian Conference on Intelligent Systems (BRACIS 2020)*, volume 12319 of *LNCS*, pages 403–417. Springer.

Tanon, T. P., Weikum, G., and Suchanek, F. (2020). Yago 4: A reason-able knowledge base. In *European Semantic Web Conference*, pages 583–596. Springer.