Language Models and Semantic Relations: a Dual Relationship

Olivier Ferret
Université Paris-Saclay, CEA, List
F-91120, Palaiseau, France
olivier.ferret@cea.fr

Abstract

Since they rely on the distributional hypothesis, static and contextual language models are closely linked to lexical semantic relations. In this paper, we exploit this link for enhancing a BERT model. More precisely, we propose to extract lexical semantic relations with two unsupervised methods, one based on a static language model, the other on a contextual model, and to inject the extracted relations into a BERT model for improving its semantic capabilities. Through various evaluations performed for English and focusing on semantic similarity at the word and sentence levels, we show the interest of this approach, allowing us to semantically enrich a BERT model without using any external semantic resource.

Keywords: Lexical semantics, lexical relation extraction, language models

1. Introduction

Language models, whether count-based or predictive (Baroni et al., 2014), and among the latter, static or contextual (Naseem et al., 2021), have a twofold relationship to semantic knowledge. On the one hand, due to their strong link with the distributional hypothesis (Harris, 1954), they have been used for a long time to extract lexical semantic relations from corpora (Lenci et al., 2022). On the other hand, many works have focused on the problem of injecting semantic knowledge into these models in order to enhance them (Wang et al., 2023b), either in the general perspective of improving how the semantic phenomena are taken into account in the tasks they are applied to, or for their adaptation to specialized domains.

The use of language models for the extraction of semantic relations is closely related to both the notions of semantic similarity (Budanitsky and Hirst, 2006) and distributional thesaurus (Grefenstette, 1994; Lin, 1998; Curran and Moens, 2002). The most common way to extract semantic relations from a language model is to rely on the ability of these models to evaluate the similarity between words on a distributional basis, an ability that is also used for their intrinsic evaluation (Faruqui et al., 2016). Applied to the vocabulary of a corpus, this capability is used to build a distributional thesaurus giving for each target word a list of distributional neighbors, ordered according to the decreasing value of their similarity, evaluated by a language model, with the target word. The first neighbors are then assumed to be the most semantically relevant, with a principled bias towards paradigmatic relations given the distributional assumption underlying the language models. Given this general principle, the main way to improve this extraction is through work at the level of the semantic similarity used to build distributional thesauri (Padró et al., 2014a,b). Nevertheless, some works also focus on improving the thesauri as such by using reordering methods, either at a global level (Claveau et al., 2014) or more locally at the level of each thesaurus entry (Ferret, 2013).

The problem of injecting semantic knowledge into language models has been the subject of a large number of studies, first focusing on static neural models and then on contextual models. Despite the differences between these two main types of models, they share the same distinction between methods operating during the building of the model and those performing the injection after its building. The latter are clearly more numerous in the case of static models, in line with Faruqui et al. (2015), while the situation is more contrasted in the case of contextual models. If we only consider the injection of lexical semantic relations,[1] the LIBERT model Lauscher et al. (2020) can be cited for the former category of methods while the latter is typically represented by the LexFit model Vulić et al. (2021).

The work presented in this paper combines the two dimensions mentioned above: it enriches a BERT-like contextual neural model (Devlin et al., 2019) by injecting lexical semantic knowledge, but unlike existing work, this knowledge is itself extracted automatically through the exploitation of neural language models. More specifically, the contributions of this work are:

- the proposal and evaluation of a new method for extracting lexical semantic relations between words based on the application of a

[1] For contextual models, existing work focuses on knowledge graphs representing factual knowledge more than on lexical semantic relations, while the trend is reversed for static models.

LREC-COLING 2024, pages 10046–10057

mask language modeling task to compounds by relying on a contextual language model such as BERT;

- the study of the complementarity of the relations extracted by this method with the relations extracted from a static language model;
- the analysis through three evaluations of the interest of the injection of such automatically extracted semantic lexical relations into a contextual language model.

2. Methods

In what follows, we first present in the next two sections two methods for extracting lexical semantic relations, one relying on a static language model while the other is based on a contextual model. In both cases, the extracted relations are expected to account for the semantic similarity between words, as opposed to the notion of semantic relatedness, as distinguished by (Budanitsky and Hirst, 2006). The union of the relations extracted by these two methods is then used as a basis for the injection of semantic knowledge into a BERT model, which is the subject of Section 2.3.

2.1. Extraction of Semantic Relations with a Static Language Model

To extract a first set of semantic similarity relations, we transpose to a static neural model the principle of selection by reciprocity in a k-nearest neighbor (k-*NN*) graph of words presented in (Claveau et al., 2014) for count-based models. More specifically, for each target word, its k nearest neighbor words are extracted based on the similarity computed between the target word and the other considered words by relying on their embeddings from a static language model, in our case a Skip-gram model (Mikolov et al., 2013a). This extraction is performed with the Faiss library (Johnson et al., 2021) using the Cosine similarity measure.[2] The distributional neighborhood relation is not symmetric by nature, but we specifically use the presence of such symmetry as a criterion for selecting the most representative neighborhood relations in terms of semantic similarity. More precisely, such a relation between words x and y is selected if y is among the k first distributional neighbors of x and conversely, if x is among the k first distributional neighbors of y.

[2]Concretely, we use the IndexFlatIP index, designed for exact search for inner product, with normalized vectors.

2.2. Extraction of Semantic Relations with a Contextual Language Model

Transposing the previous approach from static neural models to contextual neural models is much less direct than transposing it from count-based models to static neural models, in particular because a contextual model produces by definition representations of words in a particular context and not generic representations. The key point for carrying out this transposition is to be able to build a neighborhood graph of words from a language model, the neighborhood being based on the notion of semantic similarity. For a contextual model, two main strategies are possible:

- the building of static word embeddings, which brings back the problem to the first method of relation extraction presented in the previous section;
- the exploitation of the capabilities of such a model for the language modeling task used for its training.

The first strategy has already been the subject of some works (Ethayarajh, 2019; Bommasani et al., 2020; Vulić et al., 2020b; Ferret, 2022), with two main variants: one considers for a target word a set of sentences containing that word and aggregates, usually by averaging, the contextual representations produced by the language model for that word in each of these sentences.[3] The second variant builds a representation from a single occurrence of the target word in isolation, without the context of a sentence. However, Ferret (2022) shows that for building the semantic neighborhood of words, this first strategy does not give significantly better results than static embeddings, with an advantage to the first variant over the second.

Therefore, we have opted for the second strategy. We focus here on BERT-like language models, which are based on a Masked Language Modeling (MLM) task for their training. However, the approach does not exclude the use of autoregressive GPT-like models (Radford et al., 2018). The general principle is inspired by the use of BERT-like models for lexical substitution without the use of reference substitutes (Zhou et al., 2019). Nevertheless, instead of considering word occurrences in the context of sentences, we restrict ourselves to word occurrences within compounds. The application of lexical substitution to compounds can also be found in (Wang et al., 2023a) but in the context of sentences and with the fairly different

[3]Since existing contextual language models consist of several layers, the building of the representation of a word occurrence itself admits different variants.

Compound structure	Examples
ADJ NOUN	rough estimate, wearable device, motherless child
NOUN NOUN	prison guard, science academy, college student
NOUN PREP NOUN	lack of food, degree in education, return on investment

Table 1: Structure and examples of compounds considered as possible contexts for lexical substitution (ADJ : adjective; PREP: preposition).

goal of validating semantic relations between compounds that are inferred from already known semantic relations between words. In our case, the restriction to compounds is first justified by computational cost reasons, since the processing of compounds in isolation by a BERT-like model is much less expensive than the processing of sentences.[4] Moreover, experiments concerning distributional similarity with count-based models and static neural models steadily show that semantic similarity, as opposed to semantic relatedness, is better captured by a narrow context than by a broad context, which also justifies our choice of considering compounds. The analysis of attention patterns in BERT models (Clark et al., 2019) further shows that some of their heads specifically take into account these short-range interactions, suggesting that this choice is not too limiting. Finally, this method also allows, in the context of specialized domains, to exploit not only terms extracted from corpora but also terms from reference terminologies for these domains.

Concretely, the approach consists in submitting to a BERT model, with its masked word prediction layer, a set of compounds in which one of the constituents has been masked and collecting the first k predictions of the model, with their score, excluding the constituent to be predicted. Each submitted compound is processed as a separate sequence. We hypothesize that the predictions made by BERT for this compound correspond to semantic neighbors of its masked word. Applied to a large number of compounds, this method leads to the collection of semantic neighbors for a large set of words, which can be turned into a k-*NN* graph of words. Finally, as in the case of static language models, the principle of selection by reciprocity mentioned in Section 2.1 can be applied to this graph in order to select semantic similarity relations. An important point of the approach is the fact that the same word can be associated with as many neighbor lists as the number of times it appears as a masked word in a compound. To build a unique neighbor list for each word, we apply a list fusion method, more precisely the CombSum method (Fox and Shaw, 1994) with the Zero-one method (Lee, 1997; Wu et al., 2006) for normalizing prediction scores. In addition to having only one list of neighbors for each word, this fusion puts forward the predicted substitutes having recurrently the best scores and thus, ranks first the neighbors supposedly closest to the target word.

The prediction of a BERT model concerning a word to be substituted is clearly dependent on the linguistic context of this word. In our case, this context is determined by several factors: the form of the compounds in which these words appear, the role that the target words play in them, and finally, the more general context in which the compounds are placed. Regarding the first factor, since the work presented was done for English with nouns as targets, we extracted, taking as a basis the English version of Wikipedia, the compound terms including two plain words, thus obtaining the following three compound structures of Table 1, the first being about twice as frequent as the second, which is itself about twenty times more frequent than the third one.

Regarding the last factor, we have adopted the general scheme proposed by Qiang et al. (2020) in the context of lexical simplification, which consists in conditioning the sequence containing a unit to be predicted by this same sequence in its complete form. In our case, if TERM stands for the compound used as immediate context and TERM_MSK stands for the same compound with the MASK token replacing the target word, the sequence, named prompt, submitted to a BERT-like model in masked word prediction mode has thus the following form:

TERM . [SEP] TERM_MSK .

For instance in our case:

ADJ NOUN . [SEP] ADJ __ .

which can be instantiated as:

civil defense . [SEP] civil __ .
black magic . [SEP] black __ .

where __ is the location of the masked target word and [SEP] is the tag marking for BERT the separation between two sequences. This prompt is referred to as P0 in what follows.

We also tested the following variants, in particular to give a slightly broader context to compounds:

P1 this is a/an TERM . [SEP] this is a/an TERM_MSK .
P2 TERM . [SEP] this is a/an TERM_MSK .
P3 a/an TERM . [SEP] a/an TERM_MSK .
P4 TERM . [SEP] a/an TERM_MSK is a kind of TERM .
P5 TERM . [SEP] a/an TERM_MSK is a/an TERM .

[4] The attention mechanism of transformers has a quadratic complexity according to the length of sequences.

P6 TERM . [SEP] a/an TERM is a/an TERM_MSK .
P7 TERM . [SEP] a/an TERM_MSK and a/an TERM .
P8 TERM . [SEP] a/an TERM_MSK or a/an TERM .

2.3. Injection of Semantic Relations into a Contextual Language Model

For the injection of the extracted semantic relations, we relied on a contrastive approach related to metric learning. This type of approach has already been explored to perform this injection task for both static embeddings (Shah et al., 2020) and contextual language models (Vulić et al., 2021). Our work takes place in the framework defined by (Vulić et al., 2021), which itself reused the framework defined by Sentence-BERT for sentence similarity (Reimers and Gurevych, 2019). More specifically, the basic architecture of Sentence-BERT is a Siamese network exploiting a dual encoder: two sentences whose similarity is known are encoded separately by the same BERT model, a representation is built for each of the two sentences via a pooling process, and the two resulting representations are taken as input by a loss function aiming at bringing closer, through the back-propagation mechanism, the representations of the sentences known to be similar while pushing away from each other the representations of the sentences known to be dissimilar. The LexFit model of (Vulić et al., 2021) directly reuses this architecture by giving word pairs rather than sentence pairs as input and using semantic lexical relations as a reference in terms of similarity. Figure 1 illustrates our instantiation of this architecture where a pair of words (x_i, y_i) taken as input is first turned into a pair of vectors (Vx_i, Vy_i) by encoding them with a BERT model and applying an average pooling before going through a loss function. The average pooling, in that case, is used for aggregating the representations of the wordpieces of a word when it is split by the tokenizer of BERT. Many loss functions are possible to implement the general principle we have presented above, and thus indirectly inject lexical relations into a BERT-like language model; but in light of the experiments done with LexFit, we chose the Multiple Negatives Ranking (MNEG) loss (Henderson et al., 2019), which is defined as follows for a batch of B word pairs $(x_1, y_1), \ldots, (x_B, y_B))$ such that each pair (x_i, y_i) corresponds to a semantic similarity relation to inject:

$$\mathcal{L} = -\sum_{i=1}^{B} S(x_i, y_i) + \sum_{i=1}^{B} \log \sum_{j=1, j\neq i}^{B} e^{S(x_i, y_j)} \quad (1)$$

This loss function adapts the encoder's language model to maximize the similarity of each word pair (x_i, y_i) in the batch (first term of Equation 1) while minimizing the similarity of the $B-1$ artificially built pairs (x_i, y_j) (second term of Equation 1), where each y_j corresponds to the y_i of another pair in the batch, the (x_i, y_j) pairs being considered negative examples of similarity. $S(x_i, y_i)$ is the function evaluating the similarity of pairs (x_i, y_i) and (x_i, y_j).

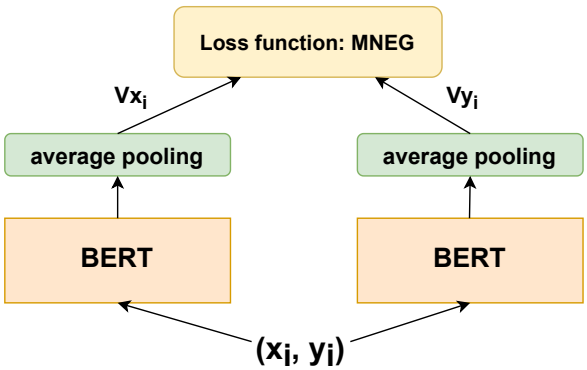

Figure 1: Architecture of our model for injecting semantic relations into a contextual language model. The weights of the two BERT models are tied.

3. Experiments

3.1. Evaluation Framework

To evaluate the results of our injection of lexical semantic relations into a contextual language model, we chose to test the capabilities of the target model in terms of semantic similarity by means of static embeddings built from this model. More precisely, our target model is the BERT *base-uncased* model and similarly to (Vulić et al., 2021), we build the embedding of a word by encoding a single occurrence of this word without any surrounding context with the target model and selecting the representation of this occurrence produced at the level of one of the 13 layers of the model (12 internal layers plus the input layer). Moreover, as in (Bommasani et al., 2020), when a word is split into several wordpieces, we build its representation by averaging the representations of its wordpieces.

The evaluation itself is based on the similarity between the representations of words produced in such a way: for each target word w_i, the set of its k nearest neighbors is selected by computing the similarity of w_i with all the other target words w_j and ordering these words according to the decreasing value of their similarity with w_i. The similarity of a pair of words is obtained by applying the *Cosine* measure to their representations. In practice, $k = 10$ and the computation of similarities relies on the Faiss library. We evaluate the relevance of this ranking by adopting classical measures of Information Retrieval, more precisely R-precision (R_{prec}), MAP (Mean Average Precision), and precisions at different ranks (P@r). Since we focus

Model	Ref.	R_{prec}	MAP	P@1	P@2	P@5	P@10
fastText-wiki	para	9.9	6.0	36.5	29.9	21.3	15.9
	syn	15.5	18.4	21.9	15.7	9.2	5.8
BERT ctxt	para	9.5	5.7	36.5	30.4	22.4	17.0
	syn	15.6	17.9	21.8	16.0	9.5	6.1
BERT iso	para	7.4	4.4	30.9	26.2	19.6	14.6
	syn	14.0	15.8	19.2	14.6	8.7	5.5
BERT refsyn	para	17.2	12.3	55.7	48.5	37.4	29.0
	syn	27.0	31.9	35.9	27.8	17.4	11.4

Table 2: Evaluation baselines and references (values $\times 100$).

globally on semantic similarity, as opposed to semantic relatedness following (Budanitsky and Hirst, 2006), our references are made of paradigmatic relations. More precisely, we consider two references, both from WordNet (Miller, 2010) since our target words are nouns in English: *para*, which gathers relations of synonymy, hyponymy, hypernymy, and cohyponymy and accounts for an extended definition of the notion of paradigmatic relation; *syn*, which is restricted to synonyms only and can be viewed as the most restrictive definition of paradigmatic relations. Our evaluations were conducted for 10,305 nouns previously used in (Ferret, 2022) and covering a wide range of frequencies.

3.2. Implementation of Proposed Methods

The implementation of the first method for extracting semantic lexical relations requires static embeddings while the second one mainly requires a set of compounds and a contextual language model. For both static embeddings and compounds, we relied on the same corpus, more precisely a dump of English Wikipedia of 10/01/2018[5] including 2.16 billion tokens, part-of-speech tagged and lemmatized using the CoreNLP tool (Manning et al., 2014) in its version 3.9.2. Compounds, limited here to bigrams of plain words, were extracted using the method defined in (Mikolov et al., 2013b),[6] with a minimum frequency of extracted terms equal to 5 and a minimum mutual information threshold equal to 0. Our static embeddings were trained following the Skip-gram model with the *word2vec*tool[7] from the lemmatized version of our Wikipedia dump.

For the implementation of the injection method, we used the Sentence Transformers library.[8] For each relation (w_1, w_2), we also considered the relation (w_2, w_1), which can be seen as a very simple form of data augmentation. Equation 1 in Section 2.3 shows that the relations to be injected are processed in batches, which have a size of 512 relations. The model is trained for 10 epochs, with a learning rate of 2e-5, the use of the AdamW optimizer (Loshchilov and Hutter, 2019), a number of warm-up steps equal to 10% of the relations to be injected, and a linear warm-up scheme.

3.3. Evaluation of Proposed Methods

Baselines and references. Table 2 gives several baselines and references according to our evaluation framework. The first of these baselines, *fastText-wiki*, corresponds to the Skip-gram model used by Vulić et al. (2021) as a reference. This static language model was trained from the English version of Wikipedia with the *fastText* tool (Bojanowski et al., 2017). The second one, *BERT ctxt*, refers to embeddings built from a contextual language model according to the method of Bommasani et al. (2020), i.e., by averaging the representations of the occurrences of target words appearing in a set of sentences. In our case, we take 10 sentences per target word and the best results are obtained with the L5 layer. As we can see, these two first references are very close, which means that concerning semantic similarity, static models and contextual models are very close as it has already been observed by other works (Lenci et al., 2022). *BERT iso* corresponds to the starting point of our process of injection of semantic relations (see Section 2.3). The best results given in Table 2 for this model are obtained from the L0 layer. Using a single out-of-context occurrence of the target words clearly hurts performance as shown by the comparison with *BERT ctxt* but requires fewer computing resources. Our last reference, *BERT refsyn*, can be considered our upper bound since it corresponds to the injection made in a BERT model by Vulić et al. (2021) of 1,023,082 semantic relations from manually built resources, more precisely WordNet and the Roget thesaurus.[9] The results are given for the L12 layer and the number of epochs is reduced to 2 given the number of relations.

Extraction of relations with a contextual model. The method we proposed in Section 2 for extracting semantic relations from a contextual model raises several questions that we try to answer here by first addressing, through the results of Table 3, the problem of the type of contextual model to adopt and the syntactic structure of the compounds used in prompts. The results are given in terms of the

[5] https://www.dropbox.com/s/cnrhd11zdtc1pic/enwiki-20181001-corpus.xml.bz2?dl=0

[6] With the implementation of Gensim: https://radimrehurek.com/gensim/models/phrases.html

[7] With the following parameters: -size 300 -window 5 -negative 10 -hs 0 -sample 1e-5 -min-count 5

[8] https://www.sbert.net/

[9] More specifically, this is our replication of the work of Vulić et al. (2021), whose code is not available.

	Para	Syn	# relations
BERT	13.9	5.2	17,007
CBERT	22.9	10.9	17,023
CBERT – heads	24.2	11.8	13,465
CBERT – modifiers	16.0	6.7	7,792
CBERT – ADJ NOUN	26.2	12.4	10,511

Table 3: Accuracy ($\times 100$) of the relations extracted by the contextual model according to the type of model and the structure of the compounds used as prompts.

accuracy of the extracted relations for our two references. Note that they were obtained with a form of prompt corresponding to P0 but with TERM and TERM_MSK being part of the same sequence (no [SEP]). The cohesion threshold of the extracted compounds by the method of (Mikolov et al., 2013b) was equal to 10. The first two lines compare the BERT *base-uncased* model with the CharacterBERT model (El Boukkouri et al., 2020), which has the same architecture as the BERT model but has the characteristic of not splitting words into subwords. This comparison illustrates the very significant impact of the splitting of words on our relation extraction task, with a very clear advantage for the CharacterBERT model, which is used hereafter.

The next three lines of Table 3 deal with the syntactic structure of compounds in prompts and the syntactic role of the target word in them. First, we note that fewer relations are produced when the target word has the role of syntactic modifier compared to the role of syntactic head. Moreover, these relations are much noisier. This observation is in line with the work of Ferret (2015), which showed that for two compounds considered semantically similar, a semantic similarity relation between their syntactic heads is more likely to exist, when they share the same modifier, than the opposite. In our case, these findings lead us to favor compounds with the target word as their syntactic head. The last line of Table 3 shows that the vast majority of relations are obtained from the term structure ADJ NOUN (NOUN being the target word), with here again less noisy relations than for the other structures. Therefore, we select only terms with the structure ADJ NOUN for TERM and TERM_MSK in what follows.

Finally, Table 4 evaluates the different types of prompts presented in Section 2, still with a cohesion threshold equal to 10 for the extraction of compounds. If most of these prompts give similar results, it should be noted that P2 and P6 obtain results that are clearly inferior to the others, without any obvious reason linked to their structure. The cause of this phenomenon is perhaps to be found in the training corpus of the model. The conclusion regarding the sensitivity of our method to the form of the prompts is therefore uncertain: if this sensitivity is not globally very strong, it can be significant for certain prompts, without a very clear explanation. Hereafter, we will retain the P0 prompt, which is the simplest one and one of the best two prompts.

	P0	P1	P2	P3	P4	P5	P6	P7	P8
Para	32.0	30.3	24.9	31.1	30.9	31.7	26.5	31.7	31.0
Syn	16.0	15.8	12.5	15.6	16.0	16.8	13.0	15.7	16.1

Table 4: Accuracy ($\times 100$) of extracted relations according to the form of the prompt.

Relations	Model	Para	Syn	# relations
Extracted	Static	30.0	19.4	35,246
	Contextual	30.6	15.6	15,473
Selected	Static	44.1	34.0	11,298
	Contextual	42.6	21.3	8,558
	Fusion	41.1	24.2	18,430

Table 5: Accuracy ($\times 100$) and number of extracted, then selected, relations.

Extraction and selection of relations: integration. The extraction of relations from a static model as described in Section 2 only requires setting the size of the neighborhood k and the target words to consider. In our case, we have chosen as targets the words with a frequency higher than 200 in Wikipedia and a value $k = 1$ for the neighborhood. Concerning the size of the neighborhood, an important difference between the two types of language models must be underlined: while the neighborhood is limited to the first neighbor for the static model, the quality of extracted relations decreasing strongly beyond that point, we extend it to the first five neighbors for the contextual model since the degradation of the quality of the relations is much more limited as the rank of neighbors increases for that type of models.

Table 5 evaluates the quality of the relations extracted by our two types of models, in comparison with their volumetry. The static model produces many more relations but the two types of models are closer in terms of quality. More precisely, they are equivalent from this viewpoint for paradigmatic relations (*para*) but the static model outperforms the contextual model for the relations of synonymy (*syn*). The selection by reciprocity in the *k-NN* graph reproduces, and even accentuates, this initial bias as far as the quality of the relations is concerned, but strongly attenuates it for the volumetry. Finally, the fusion of the two sets of relations shows their complementarity, with a small overlap between the two and performance closer to the contextual model than to the static model.

Presence in relations	Model	Ref.	R_{prec}	MAP	P@1	P@2	P@5	P@10
In-relations (4,765 words)	BERT iso	all	9.4	5.6	39.1	34.0	26.1	19.6
		syns	17.4	19.3	23.5	18.4	11.1	7.1
	Fusion	all	14.6	9.6	57.6	48.2	34.7	26.0
		syns	25.4	28.7	35.8	25.6	14.6	9.0
Out-of-relations (5,540 words)	BERT iso	all	5.7	3.4	23.8	19.5	13.9	10.4
		syns	11.2	12.7	15.4	11.4	6.7	4.2
	Fusion	all	10.0	6.2	32.6	27.2	20.4	15.6
		syns	13.8	16.7	18.9	13.9	8.5	5.5

Table 6: Split of the results of Table 7 for our initial model (BERT iso) and for our final model, after relation injection (fusion), according to the presence of the test words among the injected relations (values $\times 100$).

Model (L)	Ref.	R_{prec}	MAP	P@1	P@2	P@5	P@10
BERT iso (L0)	para	7.4	4.4	30.9	26.2	19.6	14.6
	syn	14.0	15.8	19.2	14.6	8.7	5.5
Static model (L11)	para	11.7	7.4	42.2	35.4	26.1	19.7
	syn	18.8	21.7	25.9	19.0	11.2	7.0
Contextual model (L11)	para	11.9	7.6	42.8	36.2	26.9	20.3
	syn	18.4	21.5	25.4	18.9	11.1	7.1
Fusion (L12)	para	12.1	7.8	44.2	36.9	27.0	20.4
	syn	19.2	22.2	26.7	19.3	11.3	7.1

Table 7: Results of the injection of extracted semantic relations ($\times 100$). L: best layer.

Enhancing a language model with semantic relations. The last part of this evaluation concerns the results of the injection of the extracted relations into a BERT model, illustrated by Table 7. Below our starting point, *BERT iso*, the table gives the results for each set of extracted relations: those from the static model, those from the contextual model, and the fusion of these two sets. The first observation is that the injection of relations obtains a very significant[10] performance gain compared to *BERT iso*, with particularly important improvements for all P@r measures. Globally, this gain is fairly comparable for the relations from the static model and the contextual model. It logically follows the same bias as the injected relations: it is more important in favor of synonymy for the relations extracted by the static model and larger concerning all paradigmatic relations for the relations extracted by the contextual model. The fusion of the two sets of relations leads to an additional performance gain for our two references, *para* and *syn*. The final level is of course lower than the level observed after injecting relations from manually built resources (see Table 2) but it is worth noting that in the absence of such resources, especially in the quantities used by Vulić et al. (2021), it is possible to semantically enrich a BERT-like model with automatically extracted semantic relations with a very significant performance gain.

[10]The significance of differences between *BERT iso* and the other models are judged according to a paired Wilcoxon test with $p < 0.01$.

4. Discussion and further experiments

4.1. Generalization Capabilities

The results of Table 7 show that the injection of extracted relations is globally beneficial for our evaluation set of nouns but they do not give any indication about the generalization capabilities of the injection process. More precisely, are the observed improvements limited to the words that are part of the injected relations or are they more general? Table 6 answers this question by splitting the results of Table 7 for the initial model (*BERT iso*) and the model after the injection of all extracted relations (*fusion*) according to whether the target nouns are part of the injected relations (*in-relations*) or not (*out-of-relations*).

The first observation is that the proportion of *in-relations* words and *out-of-relations* words is fairly balanced, with 46.2% for the first ones and 53.8% for the second ones. The second observation is that the improvement brought by the injection of relations is clearly not limited to the words that are part of these relations, which demonstrates the presence of a generalization effect of such an injection. This improvement is found both for the *para* and *syn* references but it is higher in terms of value and percentage for all paradigmatic relations than for synonyms with the only exception of the MAP value for *out-of-relations*. This difference is not surprising since there are more paradigmatic relations, apart from synonymy relations, than synonymy relations among injected relations as shown in Table 7. The improvement in terms of value is globally higher for *in-relations* than for *out-of-relations*, with for instance a noticeable increase of +18.5 P@1 for *para*. The trend is also observed in terms of percentage even if there are some exceptions for the MAP and R_{prec} measures in the case of *para*. These two observations illustrate the fact that while the injection process demonstrates generalization capabilities, it also favors the words that are present in the injected relations, which can be interpreted as a kind of memorization effect. However, this interpretation

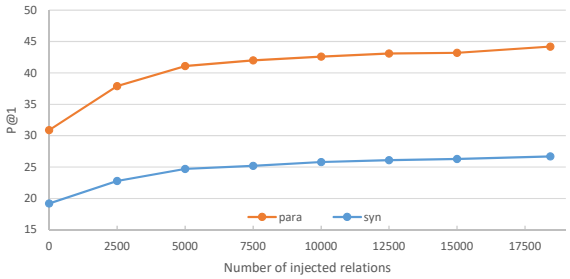

Figure 2: Impact on the number of injected relations on P@1.

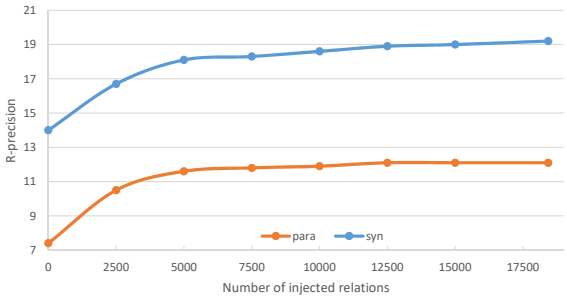

Figure 3: Impact on the number of injected relations on R-precision.

must also be modulated by another factor. The performance of the initial model for *in-relations* words is much higher than the performance of this model for *out-of-relations* words, which can be explained in terms of frequencies. On average the frequency of *in-relations* words is 3.7 times the frequency of *out-of-relations* words. The influence of this factor on semantic similarity is well-known in general but we can not discard the idea that it could also have a more specific influence on the injection process.

4.2. Influence of the Number of Injected Relations

Another way to characterize the generalization capabilities of our injection process is to study the impact of the number of injected relations on the performance of the resulting model. Figure 2 shows this impact for P@1 and Figure 3, for R-precision. To obtain these figures, we randomly selected an increasing number of relations to inject, from 0 to the total number of extracted relations with an increment of 2,500.

The global trend is similar for the two measures: while an increasing number of injected relations is steadily associated with an increasing value of these measures, the most important part of this increase is observed between 0 and 5,000 relations. The evolution beyond that point is still significant, especially for P@1 and the *para* reference, but the graph is much flatter. The presence among the injected relations of more paradigmatic relations, apart from synonymy relations, than synonymy relations is once again a plausible explanation for the better performance obtained for the *para* reference: the impact of this presence is apparently more important as the difference between the two types of relations increases in terms of number.

4.3. Intrinsic Evaluation: Word Similarity

All of our previous evaluations were focused on the characterization of the semantic relations between a word and its neighbors. Another way to evaluate the semantic capabilities of language models is to measure their agreement with human judgments about semantic similarity, which is classically done by computing the Spearman's rank correlation for a set of word pairs between human judgments and the similarity of these words computed from their embeddings by the *Cosine* measure. Table 8 shows the results of such a computation for six reference datasets (MTurk-771 (Halawi et al., 2012), Stanford Rare Word (Luong et al., 2013), HyperLex (Vulić et al., 2017), MEN (Bruni et al., 2014), Multi-SimLex (Vulić et al., 2020a), and SimLex-999 (Hill et al., 2015)) and the embeddings from our initial BERT model (*BERT iso, L0*) compared to the two BERT models with injected relations, either with reference synonyms (*BERT refsyn, L12*) or automatically extracted semantic relations (*fusion, L12*).

	BERT iso	BERT refsyn		Fusion	
MTurk 771	66.9	69.3	+2.4†	71.2	+4.3†
Stanford Rare Word	33.9	54.8	+20.9	56.6	+22.7
HyperLex	14.2	23.5	+9.3	15.4	+1.2†
MEN	65.9	68.3	+2.4†	78.6	+12.7
Multi-SimLex	51.9	72.1	+20.2	55.1	+3.2†
SimLex-999	49.7	65.0	+15.3	50.9	+1.2†

Table 8: Intrinsic evaluation of the injection of semantic relations ($\rho \times 100$). *BERT refsyn* and *Fusion* are compared to *BERT iso*. The statistical significance of differences is judged according to a two-tailed Steiger's test with $p < 0.01$.[12] † marks the non-significant differences.

As for our previous evaluation, we can observe that the injection of semantic relations, whatever their source, has a positive impact on results. This is particularly noticeable in the case of the Stanford Rare Word and illustrates the fact that the injection of semantic relations can mitigate to some extent the absence of large enough data for rare words. Interestingly, while *BERT refsyn* always outperformed *Fusion* in the evaluation of Section 3, the situation is not so clear-cut here since *Fusion* outperforms *BERT refsyn* for three datasets. One

[12] Implemented by the R package *cocor* (Diedenhofen and Musch, 2015).

Model	Pearson	Spearman
BERT iso	48.2	50.9
BERT refsyn	72.2	70.1
Fusion	71.7	69.7
(Reimers and Gurevych, 2019)		
Averaged GloVe embeddings	–	58.02
Averaged BERT embeddings	–	46.35
(Cer et al., 2017)		
Best baseline (average embeddings)	56.5	–
Lowest baseline (average embeddings)	40.6	–
Best unsupervised	75.8	–
Lowest unsupervised	59.2	–

Table 9: Sentence similarity evaluation on the STS Benchmark.

possible explanation for this result is that, similarly to our automatically extracted semantic relations, the relations underlying the pairs of words in these three datasets are more diverse than in the other datasets.

4.4. Extrinsic Evaluation: Sentence Similarity

The final evaluation we present is an extrinsic evaluation focusing on the impact of the injection of semantic relations in BERT models on the identification of semantic similarity between sentences. More precisely, this evaluation is based on the STS Benchmark dataset about semantic textual similarity (Cer et al., 2017). The principle of this task is similar to the word similarity task of the previous section but at the level of sentences: the similarity of a list of sentence pairs is computed by the model to evaluate and compared with a correlation measure against a gold standard produced by human annotators. Since our objective is to examine the impact of the injection of semantic relations and not to achieve state-of-the-art performance for this task by using training data, we adopt an unsupervised approach and more particularly, a classical baseline consisting in encoding sentences with the target model and building their representation by averaging the embeddings of their wordpieces. We use the same layers for these embeddings as in the previous evaluations. The similarity of two sentence representations is computed by the *Cosine* measure.

The upper part of Table 9 shows the results of the three models we consider while the remaining part of the table gives results from baselines coming from (Reimers and Gurevych, 2019) and (Cer et al., 2017). We can first note that as for the evaluations at the word level, the evaluation at the sentence level shows the benefit of the injection of semantic relations into BERT models, with a large difference between *BERT iso* and the two other models for our two correlation measures, Pearson and Spearman. More interestingly, *BERT refsyn* and *Fusion* are very close, with no significant difference according to a two-tailed Steiger's test ($p < 0.01$) for the two measures, meaning that for the detection of the similarity between sentences, dictionary-based semantic relations are equivalent to automatically extracted relations in terms of results. The comparison with the results reported in (Reimers and Gurevych, 2019) and (Cer et al., 2017) also illustrates the fact that baselines relying on static embeddings outperform baselines relying on embeddings built from contextual models. Finally, we can also note that the results of our enriched BERT models, not specifically designed for sentence similarity tasks, are close to the best unsupervised system reported in (Cer et al., 2017).

5. Conclusion and Perspectives

In this paper, we have presented two methods for extracting lexical semantic relations, one from a static language model, the other from a contextual model, and a method for injecting these relations to semantically enrich a contextual model. We carried out two different evaluations, one focusing on the semantic similarity at the level of words and one at the level of sentences, that have shown the effectiveness of the proposed approach, which can even compete in some cases with the injection into contextual language models of large manually built resources about semantic relations.

Besides extending our evaluations to other tasks, a possible extension of this work is to test whether, following a bootstrapping procedure, such a semantically enriched contextual language model could lead to a better extraction of semantic relations by relying on the method of Section 2.2, which in turn could lead to a better enrichment of the contextual model.

6. Acknowledgements

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council.

7. Bibliographical References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vec-

tors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4758–4781.

Elia Bruni, N Tram, Marco Baroni, et al. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Vincent Claveau, Ewa Kijak, and Olivier Ferret. 2014. Improving distributional thesauri by exploring the graph of neighbors. In *25th International Conference on Computational Linguistics (COLING 2014)*, pages 709–720.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019*, pages 4171–4186.

Birk Diedenhofen and Jochen Musch. 2015. cocor: A Comprehensive Solution for the Statistical Comparison of Correlations. *PLOS ONE*, 10(4):1–12.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *28th International Conference on Computational Linguistics (COLING 2020)*, pages 6903–6915.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *EMNLP-IJCNLP 2019*, pages 55–65.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*, pages 1606–1615.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Workshop on Evaluating Vector-Space Representations for NLP (RepEval 2016)*, pages 30–35.

Olivier Ferret. 2013. Identifying bad semantic neighbors for improving distributional thesauri. In *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 561–571.

Olivier Ferret. 2015. Early and late combinations of criteria for reranking distributional thesauri. In *ACL-IJCNLP 2015*, pages 470–476.

Olivier Ferret. 2022. Building static embeddings from contextual ones: Is it useful for building distributional thesauri? In *13th Language Resources and Evaluation Conference (LREC 2022)*, pages 2583–2590.

Edward A Fox and Joseph A Shaw. 1994. Combination of multiple searches. In *2nd Text REtrieval Conference (TREC-2)*, volume 243. NIST.

Gregory Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale Learning of Word Relatedness with Constraints. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*, pages 1406–1414. ACM.

Zellig S. Harris. 1954. Distributional Structure. *Word*, 10(2-3):146–162.

Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 5392–5404.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jeff Johnson, M. Douze, and H. Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2020. Specializing unsupervised pretraining models for word-level semantic similarity. In *28th International Conference on Computational Linguistics (COLING 2020)*, pages 1371–1383.

Joon Ho Lee. 1997. Analyses of multiple evidence combination. In *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 267—-276. ACM.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A comparative evaluation and analysis of three generations of Distributional Semantic Models. *Language Resources and Evaluation*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *ACL-COLING'98*, pages 768–774.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Seventeenth Conference on Computational Natural Language Learning (CoNLL 2013)*, pages 104–113.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL 2014: System Demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 20(5):74:1–74:35.

Muntsa Padró, Marco Idiart, Aline Villavicencio, and Carlos Ramisch. 2014a. Comparing similarity measures for distributional thesauri. In *Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2964–2971.

Muntsa Padró, Marco Idiart, Aline Villavicencio, and Carlos Ramisch. 2014b. Nothing like good old frequency: Studying context filters for distributional thesauri. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 419–424.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical Simplification with Pretrained Encoders. Technical report, arXiv preprint arXiv:1907.06226.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP 2019*, pages 3982–3992.

Sapan Shah, Sreedhar Reddy, and Pushpak Bhattacharyya. 2020. A retrofitting model for incorporating semantic relations into word embeddings. In *28th International Conference on Computational Linguistics (COLING 2020)*, pages 1292–1298.

Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart, and Anna Korhonen. 2020a. Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical fine-tuning of pretrained language models. In *ACL-IJNLP 2021*, pages 5269–5283.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020b. Probing Pretrained Language Models for Lexical Semantics. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 7222–7240.

Yizhe Wang, Béatrice Daille, and Nabil Hathout. 2023a. Exploring synonymy relation between multi-word terms in distributional semantic models. In *10th Language & Technology Conference(LTC'23)*, pages 331–336, Poznan, Poland.

Yuqi Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, Suparna De, and Amir Hussain. 2023b. Fusing external knowledge resources for natural language understanding techniques: A survey. *Information Fusion*, 92:190–204.

Shengli Wu, Fabio Crestani, and Yaxin Bi. 2006. Evaluating score normalization methods in data fusion. In *Third Asia Conference on Information Retrieval Technology (AIRS'06)*, pages 642–648. Springer-Verlag.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 3368–3373.

8. Language Resource References

Cer, Daniel and Diab, Mona and Agirre, Eneko and Lopez-Gazpio, Inigo and Specia, Lucia. 2017. *STS benchmark dataset and companion dataset*. PID http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark.

Miller, George A. 2010. *About WordNet*. Princeton University. ISLRN 379-473-059-273-1. PID https://wordnet.princeton.edu/.