Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2583–2590
Marseille, 20-25 June 2022

Building Static Embeddings from Contextual Ones: Is It Useful for Building Distributional Thesauri?

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

Abstract

While contextual language models are now dominant in the field of Natural Language Processing, the representations they build at the token level are not always suitable for all uses. In this article, we propose a new method for building word or type-level embeddings from contextual models. This method combines the generalization and the aggregation of token representations. We evaluate it for a large set of English nouns from the perspective of the building of distributional thesauri for extracting semantic similarity relations. Moreover, we analyze the differences between static embeddings and type-level embeddings according to features such as the frequency of words or the type of semantic relations these embeddings account for, showing that the properties of these two types of embeddings can be complementary and exploited for further improving distributional thesauri.

Keywords: static word embeddings, contextual word embeddings, semantic similarity, distributional thesauri

1. Introduction

The introduction of contextual language models such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) in the field of Natural Language Processing represents a major change in many dimensions. From the viewpoint of lexical semantics, one of them is the fact that these models produce representations at the token level instead of the word or type level. This change has generally a positive impact on classification or sequence labeling tasks that can be addressed by supervised machine learning approaches. However, it raises more difficulties for tasks typically addressed by unsupervised approaches and focusing on the word level, such as the extraction of semantic relations between words for instance.

One way to bypass these difficulties is to build type-level embeddings from a contextual language model[1], which was already addressed by some studies. Ethayarajh (2019) proposes to use principal component analysis (PCA) as part of his analysis of the properties of contextual models while Bommasani et al. (2020) test a larger set of operations in the perspective of using type-level embeddings for investigating the properties of contextual models. The same kind of objective can also be found in (Vulić et al., 2020b) and (Vulić et al., 2020a) with a focus on semantic properties. Finally, Chronis and Erk (2020) explore the more specific issue of multi-prototype embeddings for accounting for the diversity of token representations. To some extent, the problem we consider is also linked to the building of meta-embeddings since the problem is to combine several embeddings in both cases (Yin and Schütze, 2016; O'Neill and Bollegala, 2020).

The work of this article is more particularly focused on the production of word or type-level embeddings from contextual models for building distributional thesauri in the perspective of extracting semantic similarity relations such as synonyms. More precisely, we present three main contributions:

- first, we propose a new method for producing type-level embeddings from contextual models by introducing a kind of generalization of token representations;
- then, we perform a large-scale evaluation of this method in a complementary framework to those of Bommasani et al. (2020) or Ethayarajh (2019);
- finally, we show that considering static embeddings and type-level embeddings in a complementary perspective would be more interesting than replacing the former with the latter.

2. Method

2.1. Principles

As Bommasani et al. (2020) and Ethayarajh (2019), the method we propose starts for each word from a set of N_{tok} token representations and aims at aggregating these representations to produce a representation of that word. Each token representation corresponds to the embedding vector extracted for an occurrence of the target word in a sentence from the results of the encoding of this sentence by a contextual language model. More precisely, we distinguish three steps in the production of a representation at the type level:

- first, the selection of the considered token representations;

[1] Static embeddings such as those produced by the Skip-gram model (Mikolov et al., 2013) are intrinsically type-level embeddings but for avoiding confusion, we use in this paper the term *static embeddings* for referring to the embeddings produced by a non-contextual language model and the term *type-level embeddings* for referring to the embeddings built from a contextual language model.

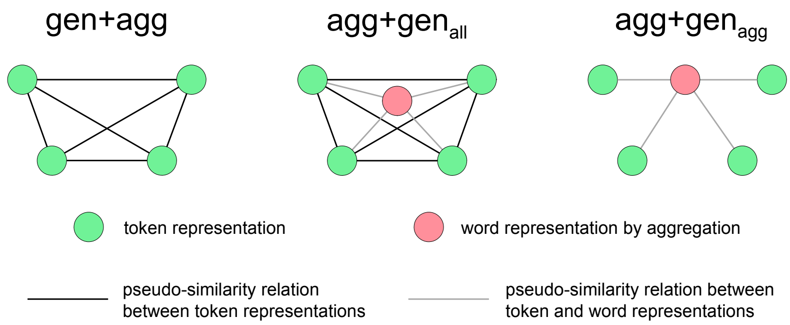
gen+agg
agg+gen$_{all}$
agg+gen$_{agg}$
token representation
word representation by aggregation
pseudo-similarity relation between token representations
pseudo-similarity relation between token and word representations

Figure 1: The three strategies for generalizing token representations. The figure graphically distinguishes the pseudo-similarity relations between token representations only and those between token and word representations but these two kinds of relations are considered similarly at the level of the retrofitting algorithm.

- then, the generalization of the selected token representations;
- finally, the building of the word representation.

The first step arises from the idea that the building of the representation of a word should benefit from the diversity of its occurrences as long as this diversity is not too large. Trying to aggregate the representations corresponding to different homonyms for instance is not a priori a good idea, which means that some sort of selection of token representations has to be done. A complementary way to control the diversity of such representations is to generalize them or at least, bring them closer. This is the objective of the second step. The last step is more directly linked to the previous work of Bommasani et al. (2020) and Ethayarajh (2019) and performs the aggregation of the selected token representations.

2.2. Token Representation Selection

The first way to restrict the diversity of the occurrences t_{ij} of a word w_i in terms of sense is to draw these occurrences from a homogeneous corpus, which we do in the experiments of Section 3. However, even if, as suggested by McCarthy et al. (2004), most words have one predominant sense in a specific corpus, it does not mean that their other senses are fully negligible. For controlling this factor and testing its influence, we assume that averaging the representations $v_{(t_{ij})}$ of the occurrences of a word should lead to a representation of this word, denoted $avg(w_i)$, very close to its predominant sense. Considering that we select a fixed number N_{sel_tok} of occurrences for each word, we propose the following options:

- *random*: it is our base option in which N_{sel_tok} tokens are randomly selected among the N_{tok} tokens initially selected for the word. This is also the option generally adopted by existing work in this area;
- *closest_avg*: we select tokens t_{ij} such that the representation $v_{t_{ij}}$ of the token is closest to $avg(w_i)$, with the idea to favor the homogeneity among the selected tokens towards the predominant sense of the word. This is a priori the best option in terms of precision;
- *farthest_avg*: this is the opposite of *closest_avg*. We select tokens such that $v_{t_{ij}}$ is farthest to $avg(w_i)$ to increase the presence of minor senses of the word;
- *uniform*: the idea is to account for the diversity of word's senses by selecting N_{sel_tok} that are uniformly distributed in terms of the similarity of $v_{t_{ij}}$ to $avg(w_i)$. This is a priori the best option in terms of recall.

2.3. Token Representation Generalization and Word Representation Building

Building a representation covering the selected tokens of a word requires to some extent erasing the differences of their representations or at least, enhancing their similarities. In concrete terms, it means bringing these representations closer to each other while keeping the core of their specificities. This objective is close to the process underlying several methods of injection of knowledge into static embeddings known under the umbrella term *retrofitting* (Faruqui et al., 2015). In the case of retrofitting methods, the process brings closer together the vectors of the words that are part of semantic similarity relations while some methods additionally push away from each other the vectors of words that are part of dissimilarity relations when they exist.

In our case, we apply a retrofitting method by considering that the token representations of a word are implicitly linked by similarity relations, which is illustrated in Figure 1. This is actually true when tokens represent different uses of the same word sense and justifies our first step of token selection. Contrary to Chronis and Erk (2020) or Wang et al. (2021), we do not cluster the occurrences of a word into distinct senses, in the case of (Chronis and Erk, 2020), or topics, in the case of (Wang et al., 2021), and do not need to introduce implicit dissimilarity relations between tokens belonging

	R$_{prec}$	**MAP**	**P@1**	**P@2**	**P@5**
CBERT-l3	15.6	18.1	21.9	15.8	9.2
CBERT-l4	15.6	18.0	22.0	15.9	9.2
CBERT-l5	14.5	16.6	20.6	14.7	8.6
CBERT-l4-all	16.1	18.4	22.5	16.3	9.4
BERTrep-l4	12.2	14.0	17.2	12.5	7.3
BERTiso-l0	14.0	15.8	19.2	14.6	8.7
BERT-l4	14.7	16.7	20.7	15.3	9.0
BERT-l5	15.6	17.9	21.8	16.0	9.5
BERT-l6	14.3	16.5	20.1	14.5	8.5
fastText	15.5	18.4	21.9	15.7	9.2

Table 1: Baseline type-level embeddings from CharacterBERT and BERT by averaging token representations.

to different clusters, either senses or topics.

While we have presented the last two steps of our method as sequential in Section 2.1, their relationship is actually more complex, with two main options. The first option, called *gen+agg*, corresponds to the scheme of Section 2.1: a pseudo-similarity relation is generated for each pair of tokens of a word, as shown in Figure 1, a retrofitting method is applied to the representations of these tokens according to the pseudo-relations and finally, the token representations are aggregated for building the type-level representation of the word[2].

The second option is partially a joint approach: the token representations are first aggregated[3]. Then, the generalization step is performed and applied to both the token representations and the result of their aggregation. The interest of this second option is to include the representation of the word into the generalization process as it is done in (Ferret, 2018) and to implement indirectly a new kind of aggregation. Two variants of this second option can be distinguished. The first variant, called *agg+gen*$_{all}$, considers the aggregation result as an additional token representation and generates pseudo-similarity relations between all the tokens, including the aggregate, as in the first option. The second variant, called *agg+gen*$_{agg}$, generates pseudo-relations only between the aggregate and all the tokens, which is a way to focus the generalization operation on the type-level representation of the word.

Similarly to Vulić et al. (2017), we adopt PARAGRAM (Wieting et al., 2015) as our default retrofitting method due to the effectiveness of its contrastive learning-alike objective function.

3. Experiments

3.1. Experimental Setup

For evaluating our method, we mainly consider two pretrained contextual language models: BERT and CharacterBERT (El Boukkouri et al., 2020), both in their `uncased` version with 12 layers (plus the input layer L0). As Bommasani et al. (2020), we build the representation of each token with BERT by averaging the representations of its wordpieces. The interest of considering CharacterBERT in our context is to investigate the impact of this representation of tokens since CharacterBERT can directly produce a representation for each token.

The building of our word or type-level embeddings is based on the encoding by these models of a set of sentences. More precisely, we randomly selected a maximal number N_{tok} of 250 sentences for each considered word w_i from the AQUAINT corpus, a 380 million-word corpus of news articles in English. For the second selection step of Section 2.2, we discarded sentences with less than 10 words and more than 90 words for having a significant and focused context when at least N_{sel_tok} ($N_{sel\_tok} = 10$) sentences fulfilled these constraints.

The evaluation itself is performed in the context of the building of distributional thesauri: for each target word w_i, a set of distributional neighbors are retrieved by computing the similarity of w_i with all the other target words w_j and ranking these words in the decreasing value of their similarity with w_i. This similarity is computed by applying the *cosine* measure to their type-level representation, built from the contextual language models. We evaluate the relevance of this ranking as in Information Retrieval with R-precision (R$_{prec.}$), MAP (Mean Average Precision), and precisions at various ranks (P@r). Similarly to work such as (Landauer and Dumais, 1997) or (Freitag et al., 2005) whose evaluation is based on the TOEFL paradigm, our reference is made up of synonyms, coming in our case from WordNet (Miller, 1990)[4], with 3 synonyms by word on average.

[2] The result of this aggregation does not appear in Figure 1 since it does not influence the generalization step.

[3] The result of this aggregation is represented by a red circle in Figure 1.

[4] More precisely, for each considered word, we gather all

	R$_{prec}$	**MAP**	**P@1**	**P@2**	**P@5**
CBERT-avg (CBERT-l4)	15.6	18.0	22.0	15.9	9.2
CBERT-pca	15.6	17.9	22.0	15.9	9.2
CBERT-gen+agg	16.1	18.6	22.6	16.3	9.5
CBERT-agg+gen$_{all}$	16.3	18.9	22.8	16.5	9.7
CBERT-agg+gen$_{agg}$	16.3	18.8	22.8	16.4	9.6
CBERT-retr-agg+gen$_{agg}$	15.6	18.0	22.0	15.9	9.2
BERT-avg (BERT-l5)	15.6	17.9	21.8	16.0	9.5
BERT-agg+gen$_{agg}$	16.2	18.8	22.5	16.1	10.1
fastText	15.5	18.4	21.9	15.7	9.2

Table 2: Evaluation of the proposed method for building type-level embeddings from contextual models.

The evaluation is performed for 10,305 nouns covering a large spectrum of frequencies.

3.2. Baseline Evaluation

The first step of our evaluations is the application of the approach of Bommasani et al. (2020), which consists in building the type-level embedding of a word by averaging the embeddings of its occurrences in a set of sentences. We present the results of this application, considered as a baseline, in Table 1 for both BERT and CharacterBERT (CBERT). More precisely, for each model, we provide the results for the best layer (shaded rows) and its two adjacent layers. These results are obtained from 10 sentences randomly selected among the 250 sentences available for each of our 10,305 nouns.

We can first observe that both models have their highest performance for nearly the same layer with very comparable values for all our evaluation measures. However, it should be noted that CharacterBERT is trained with the same settings as BERTrep, which is equivalent in terms of model and corpus size to BERT but trained with only half as many batches. The results of BERTrep in Table 1, clearly lower than BERT's results, make it difficult to conclude about the impact of averaging word-piece representations in BERT for building type-level word representations since the results of CharacterBERT may be higher with more batches.

Table 1 also provides the results of CharacterBERT without the second step of sentence selection (*CBERT-l4-all*). While the difference with *CBERT-l4* is statistically significant[5], it is not very large compared to the computational cost of taking into account all sentences, which is a little bit contradictory with the findings of Bommasani et al. (2020) in another evaluation framework. As a consequence, we will only consider the performance with the selection of 10 sentences hereafter.

Most of the approaches for building static embeddings from contextual embeddings rely on a set of example sentences containing occurrences of the target words. However, Vulić et al. (2020b) experimented with the "word in isolation" approach in which only one occurrence of each target word is encoded as a sentence by the considered contextual model and the embedding of this occurrence is used as the static embedding of the word. The line *BERTiso-l0* of Table 1 reports the performance of this approach for the best layer of BERT, which is L0 in that case. As already observed by Vulić et al. (2020b) in a different evaluation framework, this approach is outperformed by the use of several in-context occurrences for each target word. Hence, we will not consider it hereafter.

Finally, the last line of Table 1 (*fastText*) shows the results of the pretrained Skip-gram model (Mikolov et al., 2013) adopted by Vulić et al. (2020b), which was trained using fastText (Bojanowski et al., 2017)[6]. This model obtains results that are comparable to the results of the embeddings built from CharacterBERT or BERT, which is also different from the findings of Bommasani et al. (2020) and Ethayarajh (2019), made in a different evaluation context.

3.3. Evaluation of the Proposed Method

In Table 2, we first evaluate the method we propose and its different variants with the best layer of CharacterBERT (*CBERT-**) and we finally test the best variant on the best layer of BERT (*BERT-**) since the two models are close in the first evaluation. Our reference baseline is *CBERT-avg* for CharacterBERT and *BERT-avg* for BERT. All the results are obtained with 10 randomly selected sentences for each word. *CBERT-pca* corresponds to the application of PCA to token embeddings proposed by Ethayarajh (2019) instead of averaging them as Bommasani et al. (2020). Table 2 shows that the two options are equivalent in our case.

Concerning our method, we first observe that our three variants significantly outperform our reference. This improvement is not large but is sufficient to significantly outperform the Skip-gram embeddings. We also observe that our three variants are very close but that separat-

the synonyms of the synsets it is part of.

[5] The significance of differences is judged according to a paired Wilcoxon test with p equal to 0.01.

[6] https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.zip

	R$_{prec}$	**MAP**	**P@1**	**P@2**	**P@5**
random	16.3	18.8	22.8	16.4	9.6
uniform	16.4	18.9	22.9	16.7	9.7
farthest	16.5	18.9	22.9	16.6	9.8
closest	16.3	18.8	22.9	16.4	9.6

Table 3: Impact of the selection strategy of tokens for the *CBERT-agg+gen*$_{agg}$ method of Table 2.

	R$_{prec}$	**MAP**	**P@1**	**P@2**	**P@5**
unif$_{high}$	18.0	20.5	26.6	19.8	12.0
unif$_{low}$	14.8	17.3	19.3	13.4	7.3
ft$_{high}$	14.5	16.8	22.0	15.9	9.7
ft$_{low}$	16.4	20.0	21.8	15.4	8.6

Table 4: Comparison of the embeddings built from CharacterBERT (*unif*; see *uniform* in Table 3) and Skip-gram (*ft*) embeddings according to the frequency of words (*high* or *low*).

ing generalization and aggregation (*CBERT-gen+agg*) is slightly worse than joining them. The two joint variants are fairly equivalent in terms of evaluation but *CBERT-agg+gen*$_{agg}$ is computationally less intensive because of a much lower number of pseudo-similarity relations. The performance of *CBERT-retr-agg+gen*$_{agg}$, which replaces PARAGRAM by the *retrofitting* method of (Faruqui et al., 2015), confirms the interest of PARAGRAM for the task of generalization. Finally, *BERT-agg+gen*$_{agg}$ shows that the results obtained for CharacterBERT can be globally transposed to BERT.

The last aspect of the proposed method to evaluate is the strategy for selecting the tokens used for building type-level representations. This evaluation is reported in Table 3 for the *CB-agg+gen*$_{agg}$ variant, with *random* as our reference from Table 2. Contrary to the expectations arising from the fact that token representations are supposed to be strongly contextual, selecting tokens according to their proximity with their average meaning in a corpus does not have a strong influence on results. A slight advantage is observed for the strategies favoring diversity among tokens (*uniform* and *farthest*) but the results are probably limited by the fact that, as demonstrated by (Ethayarajh, 2019), the contextualization is stronger for high layers.

4. Discussion

The results we have presented in the previous section show that the method we propose for building static embeddings from contextual ones outperforms the reference methods of (Bommasani et al., 2020) and Ethayarajh (2019) in a TOEFL-like evaluation and can also be compared favorably to a method directly producing static embeddings such as fastText. In addition to these global results, we also considered more detailed results according to two dimensions.

	R$_{prec}$	**MAP**	**P@1**	**P@2**	**P@5**
fastText	15.5	18.4	21.9	15.7	9.2
uniform	16.4	18.9	22.9	16.7	9.7
Borda	17.5	20.5	24.6	17.9	10.6
RRF	17.6	20.6	24.6	18.1	10.7
CombSum	18.6	21.5	25.9	19.0	11.1

Table 5: Fusion of thesauri built from static embeddings (*fastText*) and our best type-level embeddings built from contextual embeddings (*uniform*).

Word frequency The first one is the frequency of target words. Table 4 splits the figures for both fastText's embeddings and our best type-level embeddings (*uniform* in Table 3) according to the median frequency of target words. While type-level *uniform* embeddings unquestionably outperform fastText's embeddings for high-frequency words, the trend is opposite for low-frequency words, illustrating that beyond their respective performance, native static embeddings and type-level embeddings exhibit complementary properties concerning word frequency.

This is typically an interesting configuration for applying ensemble methods. Table 5 reports the results of such an application according to a late fusion approach. More precisely, this approach, based on (Curran and Moens, 2002; Ferret, 2015), consists in fusing the thesauri built from fastText's embeddings and the type-level embeddings by merging the lists of distributional neighbors associated with each of their entries. This merge is performed according to methods used in Information Retrieval for merging ranked lists of retrieved documents. More precisely, we experimented with two kinds of methods: the *Borda* (Aslam and Montague, 2001) and *Reciprocal Rank* (RRF) (Cormack et al., 2009) fusions based on ranks and the *CombSum* fusion (Fox and Shaw, 1994) based on similarity values, normalized with the Zero-one method (Wu et al., 2006). As illustrated in Table 5, the thesauri resulting from all the fusion methods clearly outperform the two initial thesauri, which confirms the complementary nature of the two embeddings they come from. More precisely, the two rank-based methods obtain comparable results that are exceeded by the method based on similarity.

Reference semantic relations The second dimension we have considered for performing our analyses is the type of semantic relations in our gold standard. In the previous sections, we have only focused on synonyms, similarly to the TOEFL evaluation paradigm. However, the hierarchy of WordNet's synsets can be exploited for building a large set of other paradigmatic semantic relations. More precisely, we extracted three other types of relations, defined as follows:

- hypernyms [HYPE]: all the words of the synsets $\{Synset_{hype}\}$ having a direct hypernymy relation with the synsets of the target word (whose words

	$\mathbf{R}_{prec}$	MAP	P@1	P@2	P@5
SYN	16.3	18.8	22.8	16.4	9.6
	▲ 4.5%	▲ 4.4%	▲ 3.6%	▲ 3.1%	▲ 4.3%
HYPE	3.1	3.4	4.0	3.8	3.0
	▲ 6.9%	▲ 9.7%	▲ 5.3%	▲ 6.6%	▲ 7.1%
HYPO	4.2	3.2	6.6	6.4	5.5
	▲ 2.4%	▲ 3.2%	0.0%	▼ 3.4%	0.0%
COHYP	4.7	2.3	10.5	10.0	8.6
	▲ 2.2%	▲ 4.5%	▼ 0.9%	▲ 1.0%	▲ 2.4%

Table 6: Results of *CBERT-agg+gen$_{agg}$* in Table 2 according to different paradigmatic semantic relations. Below each measure, we give the percentage of difference compared to the baseline *CBERT-avg (CBERT-l4)*.

	$\mathbf{R}_{prec}$	MAP	P@1	P@2	P@5
SYN	15.5	18.4	21.9	15.7	9.2
	▼ 5.2%	▼ 2.2%	▼ 4.1%	▼ 4.4%	▼ 4.3%
HYPE	3.8	4.4	4.9	4.4	3.5
	▲ 18.4%	▲ 22.7%	▲ 18.4%	▲ 14.7%	▲ 14.3%
HYPO	4.1	3.2	6.4	6.2	5.2
	▼ 2.4%	0.0%	▼ 3.1%	▼ 3.2%	▼ 5.8%
COHYP	4.7	2.3	10.1	9.5	8.0
	0.0%	0.0%	▼ 4.0%	▼ 5.3%	▼ 7.5%

Table 7: Results of *fastText* according to different paradigmatic semantic relations. Below each measure, we give the percentage of difference compared to *CBERT-agg+gen$_{agg}$* in Table 2.

are referred to as [SYN]);

- hyponyms [HYPO]: all the words of the synsets having a direct hyponymy relation with the synsets of the target word;
- cohyponyms [COHYP]: all the words of the synsets, except $\{Synset_{hype}\}$, having a direct hyponymy relation with the $\{Synset_{hype}\}$ synsets.

We report in Table 6 the results of our best compromise between cost and performance in terms of strategy, *CBERT-agg+gen$_{agg}$* with the *random* strategy for token selection, according to the four types of paradigmatic semantic relations we consider. We first note that the best results are obtained for both synonyms (our reference type of relations in the previous sections) and cohyponyms, except for MAP in the case of cohyponyms. Even if synonyms have a clear advantage over cohyponyms, this can be regarded as a little bit surprising since synonyms correspond to the narrowest paradigmatic relations while cohyponyms correspond to the widest ones. However, this phenomenon was already observed with count-based models (Heylen et al., 2008). The values of R-precision and MAP measures are especially higher for synonyms than for cohyponyms, in part because the number of reference relations is much larger for cohyponyms than for synonyms. This is also why cohyponyms have the worst value for MAP among all semantic relations. While hypernyms obtain the worst results, we can observe that the method we propose brings the largest improvement in terms of percentage for this type of relations compared to the *average* baseline of Bommasani et al. (2020) (*CBERT-avg (CBERT-l4)* in Table 2). Conversely, while the results for hyponyms are also low, the improvement brought by our method is much lower in that case, especially for ranking them among the first neighbors. Finally, after hypernyms, synonyms are the greatest beneficiaries of our method, which suggests that it tends to favor narrower paradigmatic relations than the *average* baseline.

Table 7 focuses on the differences between fastText and *CBERT-agg+gen$_{agg}$*. We can globally note that the type-level embeddings we build from contextual embeddings outperform fastText's embeddings (the table gives the results of fastText and the percentages refer to its differences with *CBERT-agg+gen$_{agg}$*). However, hypernyms represent a strong exception in this global picture, with a large difference in favor of fastText. In the case of cohyponyms, we also observe similar values for the two types of embeddings concerning R-precision and MAP measures. More globally, the differences between fastText and our type-level embeddings are difficult to interpret in terms of types of relations since there is no correlation between synonyms and hypernyms and not even between the two best types of relations, synonyms and cohyponyms. However, as in the case of word frequency, the main lesson is that contextual embeddings do not necessarily invalidate static embeddings since they do not have exactly the same properties regarding the type of semantic relations they account for.

5. Conclusion and Perspectives

In this article, we have presented a new method for building static embeddings from contextual ones. This method is based on a threefold process starting with the selection of token representations produced by a contextual language model, their generalization, and finally, their aggregation for building type-level representations. This method was evaluated according to the TOEFL paradigm for a large number of nouns and a large number of reference synonyms from WordNet. This evaluation was performed against both reference methods for building type-level embeddings from contextual ones and static embeddings. The results of these evaluations show that our type-level embeddings outperform both other type-level embeddings and static embeddings.

Similarly to Vulić et al. (2020b), this could suggest that these static embeddings would not be useful anymore. However, our work also shows that the situation is more complex. First, type-level embeddings in our evaluations do not outperform static embeddings by a large margin while building static embeddings from a specific corpus is much easier than training a contextual language model on that corpus and building type-level embeddings from this model. More importantly, our complementary evaluations and analyses about both the frequency of words and the type of the reference semantic evaluation show that type-level and static embeddings can be complementary in tasks such as unsupervised semantic relation extraction or distributional thesaurus building since their properties are not necessarily identical.

In this work, we have focused on the building of type-level representations of words from their token-level representations but without considering the option of mixing token-level representations coming from different layers of the contextual language models. The work of Vulić et al. (2020b) suggests that associating representations coming from different layers can be interesting. We plan to study this possibility by going beyond the use of the baseline approach consisting in averaging representations, either by applying the kind of process we have proposed in this article or methods based on the projection of different representation spaces into a shared space, such as (Caciularu et al., 2021).

6. Acknowledgements

This work has been partly funded by the French National Research Agency (ANR) under the ADDICTE project (ANR-17-CE23-0001). Moreover, we would like to thank Hicham El-Boukkouri for his helpful discussions and explanations about the CharacterBERT model.

Appendix

For BERT, our experiments relied on the `bert-base-uncased` model with 110 million parameters available at: `https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip`.

For CharacterBERT and the BERTrep version of BERT, we exploited the models available at: `https://github.com/helboukkouri/character-bert#pre-trained-models`.

The encoding of sentences with the BERT and CharacterBERT models was performed on a GTX 1080 GPU card with 10GB RAM. It took 10 min for the 102,687 sentences we used for our experiments with 10 sentences by word. The other processings were performed on a CPU node of a cluster with 16 cores (Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz) and 252GB RAM. They took 3 hours, almost totally dedicated to the application of PARAGRAM.

For the generalization process, we applied both PARAGRAM and *Retrofitting* with the following parameters:

- PARAGRAM: `https://github.com/nmrksic/attract-repel`
  - number of epochs = 50
  - attract margin δ_{att} = 0.6
  - repel margin δ_{rpl} = 0.0
  - regularization constant $\lambda_{reg} = 10^{-9}$
  - batch size = 128
- *Retrofitting*: `https://github.com/mfaruqui/retrofitting`
  - number of epochs = 50

7. Bibliographical References

Aslam, J. A. and Montague, M. (2001). Models for metasearch. In *24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'01)*, pages 276—–284, New York, NY, USA. Association for Computing Machinery.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bommasani, R., Davis, K., and Cardie, C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online, July. Association for Computational Linguistics.

Caciularu, A., Dagan, I., and Goldberger, J. (2021). Denoising word embeddings by averaging in a shared space. In **SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 294–301, Online, August. Association for Computational Linguistics.

Chronis, G. and Erk, K. (2020). When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online, November. Association for Computational Linguistics.

Cormack, G. V., Clarke, C. L. A., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 758–759.

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

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

El Boukkouri, H., Ferret, O., Lavergne, T., Noji, H., Zweigenbaum, P., and Tsujii, J. (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, Barcelona, Spain (Online, dec. International Committee on Computational Linguistics.

Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 55–65, Hong Kong, China, November. Association for Computational Linguistics.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (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, Denver, Colorado.

Ferret, O. (2015). Early and late combinations of criteria for reranking distributional thesauri. In *53^{rd} Annual Meeting of the Association for Computational Linguistics and 7^{th} International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), short paper session*, pages 470–476, Beijing, China, July.

Ferret, O. (2018). Using pseudo-senses for improving the extraction of synonyms from word embeddings. In *56^{th} Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 351–357, Melbourne, Australia. Association for Computational Linguistics.

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

Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., and Wang, Z. (2005). New experiments in distributional representations of synonymy. In *Ninth Conference on Computational Natural Language Learning (CoNLL 2005)*, pages 25–32, Ann Arbor, Michigan, USA.

Heylen, K., Peirsman, Y., Geeraerts, D., and Speelman, D. (2008). Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In *Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding Predominant Word Senses in Untagged Text. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 279–286, Barcelona, Spain.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR 2013, workshop track*.

O'Neill, J. and Bollegala, D. (2020). Meta-embedding as auxiliary task regularization. In *ECAI*, pages 2124–2131. IOS Press.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 2227–2237, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Vulić, I., Mrkšić, N., Reichart, R., Ó Séaghdha, D., Young, S., and Korhonen, A. (2017). Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules. In *55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 56–68, Vancouver, Canada, July.

Vulić, I., Baker, S., Ponti, E. M., Petti, U., Leviant, I., Wing, K., Majewska, O., Bar, E., Malone, M., Poibeau, T., Reichart, R., and Korhonen, A. (2020a). Multi-SimLex: A Large-Scale Evaluation of Multilingual and Crosslingual Lexical Semantic Similarity. *Computational Linguistics*, 46(4):847–897, 02.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., and Korhonen, A. (2020b). Probing Pretrained Language Models for Lexical Semantics. In *2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 7222–7240, Online. Association for Computational Linguistics.

Wang, Y., Bouraoui, Z., Espinosa Anke, L., and Schockaert, S. (2021). Deriving Word Vectors from Contextualized Language Models using Topic-Aware Mention Selection. In *6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 185–194, Online, August. Association for Computational Linguistics.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2015). From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

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

Yin, W. and Schütze, H. (2016). Learning word meta-embeddings. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1351–1360.

8. Language Resource References

Miller, G. A. (1990). WordNet: An On-Line Lexical Database. *International Journal of Lexicography*, 3(4).