Using Syntax for the Semantic Representation of Sentences

Iskandar Boucharenc †[1], Eve Sauvage †[1,2], Thomas Gerald [1], Julien Tourille [2], Sabrina Campano [2], Cyril Grouin [1], Sophie Rosset [1]
[1]LISN, 507 rue du Belvédère, 91405 Orsay,
{name.surname}@lisn.fr
[2]EDF R&D, 7 Bd Gaspard Monge, 91120 Palaiseau
{name.surname}@edf.fr

Abstract

Deep learning methods in natural language processing often rely on statistical methods to tokenize texts before vectorization. This segmentation produces lexical subunits offering great flexibility. However, the reuse of identical tokens across words with different meanings can favor representations based on surface form rather than on linguistic information, especially semantics. This mismatch between semantics and surface form can lead to undesirable effects in language processing. To limit the influence of form on the semantics of vector representations, we propose an intermediate representation based on syntactic parsing that is more compact and more faithful to word meaning.

Keywords: Deep learning, Pre-Trained Models, Latent Representation, Constituent Analysis, Tokenization

1. Introduction

Deep learning methods for Natural Language Processing (NLP) rely on text vectorization to compute refined representations. The most commonly used method involves training statistical segmentation algorithms (tokenizers), such as SentencePiece (Kudo & Richardson, 2018), to break text into subunits that models can process. This tokenization results in subword-level segmentation, producing modular tokens that avoid errors caused by out-of-vocabulary words.

However, Tytgat et al. (2024) shows that deep models tend to favor the surface form of the lexicon over semantics in textual similarity computation. This preference for the surface form can lead to poor performance in tasks requiring fine-grained distinctions in sentence meaning, such as in Semantic Textual Similarity (STS) tasks or paraphrase detection (Muennighoff et al., 2023).

To address this issue, our hypothesis is to take advantage of structured linguistic data to make vector representation more faithful to the semantics of their text. We propose to improve the semantic information contained in lexical embeddings by using syntax which is closely related to semantics but benefits from larger parsing models. In this article, we explore three hypotheses: (i) the predominance of surface shape over semantics is linked to a lack of corelation between statistical segmentation and semantic information; (ii) Semantics-aware segmentation using constituency parsing could reduce the dependence of vector representations on surface form; finally, (iii) the merging of tokens representations belonging to the same constituent could help disambiguate vector representations. Exploring these hypotheses, we demonstrate that (a) models

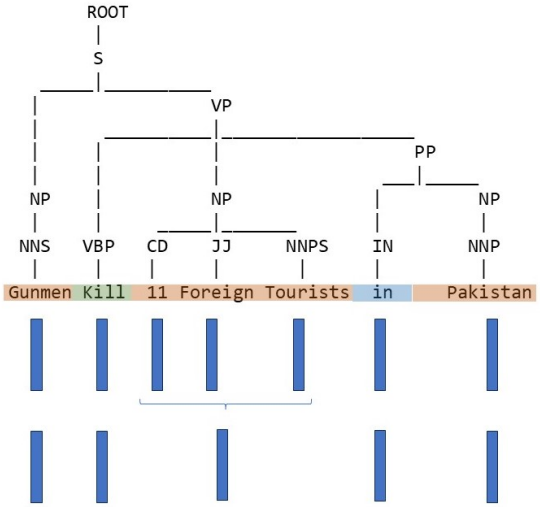


Figure 1: Example of token merging on a text from the *STSbenchmark* corpus. The different phrases are represented using color groups. The blue bars are embedded representation. Top blue bars represent the original token segmentation, while bottom blue bars represent the merged version.

are able to retain around 95% of their performances with deeply transformed representations; (b) Merging representations reduces the input sequence lengths from 26% to 43% in terms of number of tokens, depending on datasets; and (c) In the case of BERT, thanks to the length reduction due to merging token representations, a speed-up of 27.63% was achieved on average on the Semantic Textual Similarity (STS) subset of MTEB (Muennighoff et al., 2023).

The remainder of this article is organized as fol-

† These authors contributed equally

Proceedings of the Workshop on Structured Linguistic Data and Evaluation, pages 169–179

lows: section 2 presents the work related to our approach and our motivations. Section 3 details the methods used and the experimental protocol. Finally, we present our results and analyses in section 4, before concluding in section 5.

2. Related Works

In Natural Language Processing (NLP), methods derived from deep learning (DL) now dominate most tasks. These methods require representing text as vectors. The first vectorization approaches relied on one-hot encoding, a simple method that suffered from two major limitations: first, it produced very high-dimensional vectors; second, it did not account for the context in which words appeared.

With the rise of artificial neural networks, more efficient solutions have emerged, allowing to take context into vector representations through unsupervised learning (Mikolov et al., 2013; Pennington et al., 2014). These approaches exploit tasks such as masked language modeling (e.g., fill-in-the-blank text) to extract contextualized representations. However, while these methods improve context modeling, they still struggle to effectively capture the semantics of rare or out-of-vocabulary words. To overcome this limitation, Sennrich et al. (2016) proposed subword, or token, segmentation based on statistical rules, a technique for encoding words absent from the training vocabulary, albeit imperfectly.

However, this segmentation is based solely on statistical criteria, without linguistic grounding, which can introduce a bias in textual representation linked with surface form (Tytgat et al., 2024). Currently, the most efficient DL models are based on the Transformer architecture (Vaswani et al., 2017). Despite being highly parallelizable, these models exhibit quadratic complexity with respect to the length of the sequences processed. Therefore, subword segmentation poses an additional problem in terms of resource consumption, both computational and memory-wise (Dettmers et al., 2023).

Several recent studies suggest that the vocabulary size of a model follows a scaling law (Tao et al., 2024; Huang et al., 2025). Their experiments indicate that increasing vocabulary size could benefit the performance of models with a large number of parameters. Indeed, Tao et al. (2024) point out that the optimal vocabulary size is often underestimated, while Huang et al. (2025) demonstrates that vocabulary adaptation can be performed flexibly enough to decouple the input vocabulary from the output vocabulary. Even though merging token vectors does not change the vocabulary of the model, the model still has to learn to use new representations from its embedding layer.

Although the limitations of subunits are well known, the literature has mainly focused on model optimization (Dao et al., 2022; Dettmers et al., 2023). Nevertheless, some studies have explored input compression strategies to improve model efficiency. For example, in computer vision (CV), the visual token merging (TokenMerging (ToME)) approach (Bolya et al., 2023) has proven effective in speeding up processing. In NLP, Gee et al. (2023) studied the impact of increasing the vocabulary of language models by adding polylexical expressions, based on the frequency of n-gram occurrence. Our work follows this approach, but differs by the use of a non-statistical merging criterion based on linguistic information. Approaches based on a frequency criterion can, indeed, lead to performance losses on certain tasks (Kumar & Thawani, 2022).

3. Methodology

The aim of this study is to examine the impact of syntactic-based modifications of the token representations on encoder models. This requires adapting the model weights through a specific training process. This section details motivation for the use of syntactic parsing (3.1), the criteria and motivations guiding the representation modifications (3.2), the selected models and datasets (3.3), and the methodological choices adopted for model training (3.4).

3.1. Motivation

Syntax and semantics are two alternative views of one linguistic entity (namely its form and its meaning) with syntax providing the necessary means to assign meaning to utterances. In NLP tasks focusing on semantics, syntax is often used to determine boundaries. For example, Semantic Role Labeling (SRL), Named Entity Recognition (NER), Open Information Extraction (OIE) or Abstract Meaning Representation tasks can be enhanced by means of syntactic information (Cao & Clark, 2019; Dong et al., 2022; Huang et al., 2023). We based our study on these parallels and especially on the semantic aspect of constituents, also used in SRL.

Syntactic analysis has already been used with Large Language Models (Xu et al., 2021; Shen et al., 2021) to improve their understanding of language.

Reduction of the Impact of Surface Form The main motivation behind token merging lies in the predominant consideration of surface form in text encoding. A word or group of words belonging to a coherent semantic unit can be segmented into several tokens (see Figures 1 & 3), while words

with distinct meanings may share identical tokens. This artificial proximity in the representation space can lead to an overestimated similarity between semantically distant sentences. To address this problem, we propose an intermediate representation that mitigates the impact of surface form by merging tokens belonging to the same semantic unit (cf. figure 1).

With this method, we only increase the vocabulary size artificially by adding an intermediate representation after the embedding phase. Thus, by integrating a merging step into the model, we seek to evaluate its impact on performance and input data compression.

Input Compression Merging tokens in a text reduces the number of tokens to be processed by the model and, consequently, reduces the length of the sequences. Since Transformer models have quadratic complexity with respect to sequence length, this reduction represents a significant computational advantage.

3.2. Modifiying Token Representations

To test our hypotheses, we propose an approach to merge the representations of certain tokens based on syntactic analysis. This section presents the method used, the criteria selected, and the implementation details.

Merging Criteria Given the importance of semantic representation in language models and the semantic lens adopted by Mikolov et al. (2013) or Peters et al. (2018) to explain lexical embeddings, it seems appropriate to adopt a merging criterion based on semantics. A division into lexemes, the smallest units of meaning, might seem like a logical approach. However, there are neither reference corpora nor readily available lexeme segmentation models. Moreover, such an approach would not allow the consideration of MultiWord Expressions (MWE) and would require a profound modification of the representations used by the models.

We therefore propose an approach based on constituent analysis rather than phrase parsing for its greater versatility. This analysis allows us to extract phrases from different sentences on different levels, which are often considered independent semantic units as well as syntactic units. They are identifiable using constituent tests that highlight their structural autonomy (topicalization, substitution, dislocation, splitting, etc.). Constituent analysis makes it possible to identify and reconstruct independent semantic units within a sentence, providing a clearer representation of its internal structure by incorporating additional information that is not captured by dependency analysis alone. Indeed, dependency analysis lacks information on membership in grammatical categories, as theorized by Chomsky (1956). Unlike an approach limited solely to named entities, constituent segmentation offers broader coverage across the entire text. Although this method can identify a large number of embeddings to merge, it results in a substantial modification of the distribution of vectors, which could be detrimental to the performance of the model.

We choose to base the merger on the noun (NP), prepositional (PP), and verbal (VP) phrases closest to the leaves of the syntax tree. This selection allows us to obtain constituents of significant size, thus significantly reducing the length of the models' input sequences. For the sake of simplification, we exclude adverbial and adjectival phrases, even though they are among the five generally identified types of phrases. Indeed, these phrases are often included in other phrases, allowing us to consider them without processing them separately.

Implementation Details For constituent analysis, models are provided by Stanza (Qi et al., 2020), one of the few analyzers available for this task. Once constituents are identified, their boundaries are aligned with the output of the tokenizer. The obtained alignment is used to build merging matrices that send the original representation to the new one. The matrix is composed of 1 for unchanged representation, $\frac{1}{t}$ for constituents formed by t tokens to merge, 0 elsewhere. The remainder of the process is performed using matrix operations from the PyTorch library, enabling fast batched merging.

3.3. Tasks, Datasets and Models

Tasks The relevance of the obtained representations is evaluated along three criteria. First, training and evaluating models on the STSBenchmark dataset, from the Massive Textual Embedding Benchmark (MTEB) (Muennighoff et al., 2023), with and without merged representations to evaluate if the new representations are learnable. Second, after training, models and representations are evaluated on the test sets of the other STS tasks from MTEB to assess their robustness and generalizability. Third, an evaluation is conducted following the work of Tytgat et al. (2024) to gauge whether the surface form is less predominant after merging. The main purpose of this evaluation in our experimental setting is to verify that the representation of a synonym for a term is closer to that of the term in question than that of a near form (homophone, homograph or cognate), which we call paronyms. We can argue that using paronyms instead of homographs to evaluate the impact of similar tokens' apparition on sentence similarity is under effective.

We use the corpus proposed by Tytgat et al. (2024), which we will refer to as the SYNPAR corpus. We conduct our experiments in English. All datasets are presented in Table 1.

Training Datasets The models are trained during two phases, a pretraining phase on a subset of Wikipedia and a finetuning phase on STSBenchmark.

1. Pre-training on a subset uniformly sampled from Wikipedia EN[1] due to the richness and diversity of the topics covered, making this corpus a suitable basis for initial model training.
2. Fine-tuning on MTEB/STSBenchmark. We only use the train split for the finetuning in order to avoid data contamination.

Models We use the models studied by Tytgat et al. (2024), namely BERT (Devlin et al., 2019)[2] and all-miniLM-L6-v2[3]. The choice of these models is based on several criteria:

- their lightweight nature, which facilitates the adaptation and reproducibility of experiments
- the availability and transparency of their training data
- their prior training on limited amounts of data, thus limiting the impact of acquired biases

While `all-miniLM-L6-v2` is adapted to text similarity tasks, `bert-base-uncased` offers an optimal basis for experiments because of its task agnostic pretraining.

3.4. Model Training

To allow the models to adapt to the merged representation, we modify their weights. Our approach is based on two steps: (i) pre-training based on a task analogous to MLM (*Masked Language Modeling*) and (ii) fine-tuning on a sentence similarity task, as summarized in table 3.

Pre-training Merging representations automatically results in a change in the length distribution of representation vectors (see Figure 2), reducing the length of sequences by 42.83% on average in the Wikipedia corpus. To mitigate the impact of this change, a common approach (applied, for example, during a change of domain) is to continue the pre-training of the model. Given that the models used were initially pre-trained on an MLM task, we adapt this task to our representation in the new context.

This task adaptation is necessary because the merged representations no longer directly correspond to vocabulary tokens, making the traditional objective of the pre-training task using cross-entropy on tokens inapplicable. We therefore opt for a continuous approach to this task (*i.e.* the token vector representation is masked instead of its index). The objective of this pre-training is to allow the model to acquire a semantic understanding of the merged token groups. Rather than optimizing cross-entropy between the prediction probabilities of masked tokens, we train the model to reconstruct the vector resulting from merging before passing it into the network: the representations are first merged, then the resulting groups are masked to be reconstructed by the model. The representation produced by the final layer is processed by a perceptron, then compared to the corresponding input vector. The L^2 distance between the output of the perceptron for a masked representation and the merged representation is retained as the loss function. This is summarized below.

$$\mathcal{L}(X, \tilde{X}) = \mathrm{MSE}(X, \mathcal{F}(X))$$

with $\mathcal{F} = p \circ \mathcal{M}$ where p is a perceptron whose input and output dimensions are the dimensions of the model $\mathcal{M}$, $\tilde{X} = \mathcal{F}(X)$. The model and perceptron are trained on a randomly selected subset of Wikipedia representing 14M tokens. We save the model producing the best average loss on 3200 texts.

Adaptation In parallel with pre-training the models to understand merged tokens, we propose adapting these models for the sentence similarity task by incorporating the merging step. Unlike the masked token task, the model's loss can be calculated similarly to the training of the Siamese networks proposed by Reimers & Gurevych (2019). More precisely, we evaluate the cosine similarity between the sentence representations of a pair of sentences after applying the merging procedure. Then, the MSE (Mean Squared Error) loss is calculated between the ground-truth similarity and the similarity predicted by our approach. We use the Spearman correlation (r_s) on the validation set as a stopping criterion.

3.5. Evaluation

We use the SynPar dataset to evaluate the semantic expressiveness of our representations. Indeed, this dataset emphasises the contrast between lexical forms and meanings, providing a good founda-

[1] https://huggingface.co/datasets/legacy-datasets/wikipedia

[2] https://huggingface.co/google-bert/bert-base-uncased

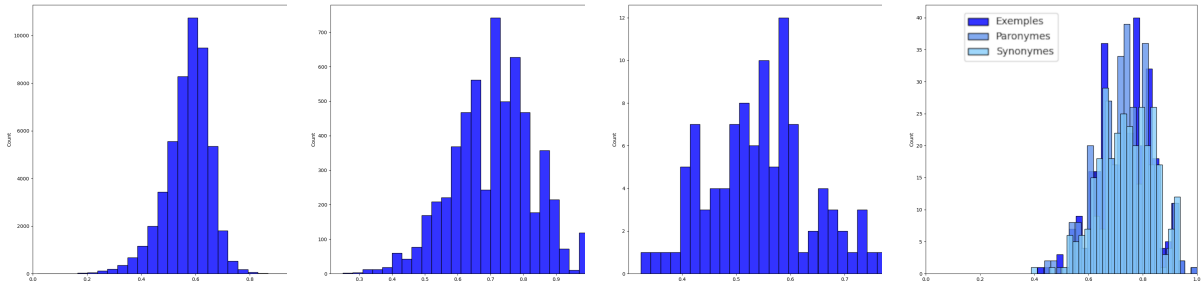
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Dataset	Description	# Test Examples
SynPar	Sentence sets in which one of the words is derived into four variants: original, synonym, paronym (cf. table 2)	372
STSBench.	Compilation of various pair of sentences for Semantic Similarity	5.75k
BIOSSES	Semantic sentence similarity estimation system for the biomedical domain	100
SICKR	Sentences Involving Compositional Knowldedge	9.93k
STS12	Semantic similarity from existing paraphrase datasets and machine translation evaluation resources	2.23k
STS13	Typed Semantic Similarity	1.5k
STS14	Emotion labelled corpus of tweets	3.75k
STS15	Semantic Similarity with Referential Translation Machines	3k
STS16	Sentiment Analysis in Twitter	1.19k
SummEval	Evaluation of machine generated summaries	100

Table 1: Summary of the different tasks evaluated. Futher information can be found in the MTEBenchmark paper (Muennighoff et al., 2023)

original	paronym	synonym
Please **accept** this gift as a sign of my gratitude.	Please **except** this gift as a sign of my gratitude.	Please **receive** this gift as a sign of my gratitude.

Table 2: SynPar dataset exemple



(a) Wikipedia (b) STSBenchmark (c) BIOSSES (d) SynPar

Figure 2: Size of the merged texts compared to the original texts. The size ratio is read on the x-axis (percentage), and the number of sequences with this ratio is read on the y-axis.

Model	conf.	merged	pre-trained	finetuned
name	zs	✗ / ✓	✗	✗
	pt	✓	✓	✗
	ft	✗ / ✓	✗	✓
	pt-ft	✓	✓	✓

Table 3: Summary of the experiments carried out. Pre-training (pt) is done on 50,000 Wikipedia texts, and fine-tuning (ft) is done on the STSBenchmark game. The abbreviation zs stands for zero-shot evaluation. We will use these terms in the rest of the article for clarity.

tion for semantic evaluation. We verify the learnability of merged representations with an ablation study on STSBenchmark. In this study, we compare pretraining, fine-tuning, and zero-shot settings. The robustness of the performances is evaluated on the other MTEB STS datasets.

ABX Test In order to compare our results with those of Tytgat et al. (2024), we use the ABX test described by Carlin et al. (2011) and Schartz et al. (2013). This test measures, between 0 and 100, the similarity between an element A and an element B having minimal differences with it, in comparison with an element X, which has major differences (equation 1).

$$100 \times \frac{\text{Card}(\cos(A,B) \geq \cos(A,X))}{\#\text{Examples}} \quad (1)$$

Because we need to obtain one vector per sentences, we have to agregate the intermediate representations obtained. We experiment with three

different agregations configuration : using the sentence's mean representation, using the pooler of the model representations or using only the CLS of the sentence.

Spearman Correlation As in the evaluation of the MTEB Benchmark, we use the Spearman (1904) correlation r_s for the evaluation on this corpus. We exclude the Pearson correlation, following the recommendations of Reimers et al. (2016).

4. Results & Analyses

4.1. Results on STSBenchmark

To evaluate the results on the training corpus, as in the literature, we calculated the Spearman correlation for the different models whose results are reported in table 4 using the notations defined in table 3. The table is divided into two parts, one for each base model. The first two rows represent the model without constituent merging. The second row corresponds to a fine-tuning without merging as a control. The following three rows can be interpreted as an ablation study by successively adding fine-tuning (ft) and pre-training (pt-ft) phases.

		merge	r_s validation	r_s test	Δ (test)
bert	zs	✗	59.32	47.29	–
	ft	✗	**87.48**	**84.80**	**+37.60**
	zs	✓	48.46	40.13	-7.16
	ft	✓	83.96	79.05	+31.76
	pt-ft	✓	84.21	79.65	+32.36
miniLM	zs	✗	86.72	82.03	–
	ft	✗	**89.51**	**85.49**	**+3.46**
	zs	✓	59.24	42.81	-39.22
	ft	✓	85.05	78.10	-3.93
	pt-ft	✓	86.08	81.03	-1.0

Table 4: Spearman correlation (r_s) for the different models and performance difference with the zero-shot evaluation (Δ).Results are grayed out when the inference representation does not match the training representation of the models (original model on merged representation or vice versa). The best scores are in bold, and the second bests are underlined. *cf.* table 3 for the explanation of the model names.

Maintaining the zero-shot framework is no longer feasible due to changes in the representation distribution, as evidenced by the significant drops in performance for the zs evaluation on merged representations. In the case of the BERT model, merging yields good results, even when fine-tuning the model without merging. The increase in the correlation metric with the compression ratio of 29.91% validates our assumptions for the BERT model. However, in the case of MiniLM, any merging attempt results in a drop in performance. Although pre-training puts this drop into perspective and the average compression ratio remains attractive, it is more difficult to assess this model. Note that the objectives of the two models differ slightly. The all-miniLM-L6-v2 version is designed to generate efficient general-purpose embeddings and has a more complex pretraining procedure, which could explain the difficulty in recovering from the drop in performance. BERT, on the other hand, is intended to be fine-tuned on downstream tasks.

4.2. Results on Other MTEB Tasks

Further evaluations are conducted on other MTEB tasks. Firstly, Table 5 compares results on the remaining STS tasks of MTEB. All models are fine-tuned on the STSBenchmark training set and evaluated on their respective test sets. Average score and inference times are reported in the last two columns. Both models experience a slight drop in performance but retain 94.45% (BERT) and 96.34% (MiniLM) of the unmerge counterpart. The major drop in performance is due to BIOSSES. Indeed, the analysis of relative sequence lengths shows that, due to the presence of scientific terms, the text is overtokenized, as shown in Figure 3. The same effects, but to a lesser extent, occur with STS16, which draws examples from Twitter. Regarding inference time, while BERT shows a 27.63% speed-up on the STS subset of MTEB on average, MiniLM shows a 19.56% slowdown, with $0.0088\,\mathrm{s}$ more per example on average. Secondly, Table 6 compares the results obtained by BERT and MiniLM on the Summeval dataset. Another difference between the two models is that Fine-Tuning BERT with token merging on STSBenchmark improves its results, while MiniLM suffers from this tuning. Regarding inference time, BERT still benefits from a speed-up, as does MiniLM this time.

4.3. Results on the SynPar Corpus

For SynPar dataset, we find results similar to those in the previous section on table 7: the gains with BERT are much greater. While MiniLM shows a decrease in ABX scores, BERT outperforms MiniLM when average of the vectors is taken as a representation (see BERT-PT-FT in the table 7). The 2.6-point improvement in score over the zero-shot version confirms that merging representations is beneficial for the BERT model, both for compressing sequences and for avoiding confusion due to paronymy. On MiniLM, an improvement is visible on unmerged data after pretraining the model on merging for `cls` configuration. There is, therefore, room for improvement. Nonetheless, it shows the

	merge	STSB	BIO.	SIC.	STS13	STS14	STS15	STS16	STS12	Avg.	Ratio	t.(sec)
bert	✗	**84.60**	**72.73**	**73.76**	**87.59**	**87.78**	86.95	**81.03**	**78.26**	**81.58**	–	0.154
	✓	79.65	57.73	72.02	84.83	85.67	**87.06**	75.82	73.83	77.05	94.45	**0,111**
miniLM	✗	**85.69**	**83.15**	**77.82**	**88.25**	85.98	**89.68**	**82.85**	**75.53**	**83.61**	–	**0.045**
	✓	81.03	66.06	73.67	86.21	**86.68**	88.59	79.95	73.83	79.53	95.12	0.054

Table 5: Spearmanr results on the STS datasets. Models are finetuned on STSBenchmark and then evaluated on the remaining data. The average processing times are reported in the last column. BIOSSES and SICKR are abbreviated BIO. and SIC. respectively

		merge	r_s	t.(sec)
bert	zs	✗	29.82	–
		✓	15.37	2.11
	ft	✗	27.79	2.40
		✓	**32.49**	**1.29**
miniLM	zs	✗	**30.81**	–
		✓	25.77	**1.10**
	ft	✗	25.33	1.14
		✓	23.13	1.27

Table 6: Results on SumEval dataset comparing with (✗) and without (✓) merging tokens and, the zero-shot (zs) approach versus the fine-tuned (ft) configuration on STSBenchmark.

		merge	configurations		
			mean	pooler	cls
bert	zs	✗	66,88	56,17	66,56
		✓	60,39	55,84	58,44
	pt	✗	62,01	64,61	65,26
		✓	61,04	57,80	57,80
	pt-ft	✗	67,53	63,64	**67,21**
		✓	**69,48**	**65,26**	66,23
miniLM	zs	✗	**67,53**	**68,83**	68,83
		✓	59,74	64,29	63,31
	pt	✗	65,91	67,53	**69,16**
		✓	58,17	62,99	62,99
	pt-ft	✗	64,29	65,91	66,88
		✓	58,77	62,66	64,94

Table 7: Results of ABX tests on the SYNPAR corpus comparing the addition of token merging. Results are grayed out when the inference representation does not match the training representation of the models (original model on fused representation or vice versa). The best scores are in bold.

particular nature of MiniLM embeddings and their over-optimisation.

Two interesting points emerge from these experiments : when only the embedding layer is considered, merging representations improves ABX scores from 52.6% to 56.82% for `bert-base-uncased` and from 49.02% to 50.32% for `miniLM`. The second point is that the overall best result for `MiniLM` happens for PT without merged representation on CLS. This result is surprising because of the inadequacy between the pre-trained representation and the model inference representation. The best BERT CLS ABX score also occurs on inadequate data. It could mean that model learning merged representation may help even for unmerged representations on CLS, which is only indirectly affected by the merge.

4.4. Discussion

Based on the observation that language models calculate higher similarity with paronyms than with synonyms, we proposed a method to merge tokens based on linguistic criteria. The evaluation took three criteria into account: the first, performance on a common NLP task, STSBenchmark. The second, robustness and generalization to unseen data. The third, more qualitative, focused on the distinction between paronyms and synonyms in the ABX test. In both cases, the method proved conclusive after a pre-training and fine-tuning phase on the STS task. Considering the evaluation on the STS-Benchmark dataset, the pre-trained and fine-tuned BERT model yields the best results with a compression ratio of 29.91%. MiniLM shows a 1-point drop in performance but retains the compression property. In both cases, the experiments demonstrated fairly good stability to changes in learning rate.

In contrast to our initial assumption (the improvement of sentence similarity performance with merged intermediate representations), the qualitative evaluation was conclusive only for BERT. Indeed, these results show that the STS task helps distinguish paronyms in this model, as fine-tuning improves performance. In contrast, MiniLM shows a decrease in the ABX score, except for the pt-ft version evaluated without merging. This result may be symptomatic of a poor training method. We assume that improvements could be achieved by pretraining on more data, which currently contains only 14 million tokens. Note that our results on STS cannot be considered an improvement in the task since, unlike the original models, we are moving away from the zero-shot framework. Thus, merging representations achieves results that retain 94% of the score of the original models while reducing the length of the input sequences significantly.

Original: "T47D, MCF-7, Skbr3, HeLa, and Caco-2 cells were transfected by electroporation as described previously."

Tokenized: ‘t’, ‘##47’, ‘##d’, ‘,’, ‘mc’, ‘##f’, ‘-’, ‘7’, ‘,’, ‘sk’, ‘##br’, ‘##3’, ‘,’, ‘he’, ‘##la’, ‘,’, ‘and’, ‘ca’, ‘##co’, ‘-’, ‘2’, ‘cells’, ‘were’, ‘trans’, ‘##fect’, ‘##ed’, ‘by’, ‘electro’, ‘##por’, ‘##ation’, ‘as’, ‘described’, ‘previously’, ‘.’

Merged: ‘t47d’,‘,’, ‘mcf-7’,‘,’, ‘skbr3’,‘,’, ‘hela’, ‘and caco-2 cells’, ‘were transfected’, ‘by electroportation’, ‘as described previously’

Figure 3: Original, tokenized, and merged processing of a BIOSSES Benchmark example.

Merging tokens based on a constituent analysis criterion allows, across the four datasets, an average reduction of 42.83%, 29.91%, 45.66%, and 26.42%, respectively, in the size of Wikipedia, STS-Benchmark, BIOSSES, and SynPar. More precise distribution visualization can be observed in figure 2, which shows the histograms of the relative sizes after merging. As explained in section 3, this reduction therefore allows for longer contexts to be taken as input without increasing the complexity of the models.

Inference Acceleration

As suspected, the merging procedure generally lacks generalizability. This drawback is salient on the domain-specific BIOSSES dataset. However, reducing sequence lengths automatically consequently reduces inference time. On average, an increase in inference speed of 27.63% is observed with BERT. The highest increase in speed is obtained on the BIOSSES dataset, which shows a significant reduction in sequence lengths (see Figure 2d). It should be noted, however, that this comes with a preprocessing time that can be costly, since the constituent parser is quite slow. Preprocessing time ranges from $0.11\,\mathrm{s}$ to $0.24\,\mathrm{s}$. Possible solutions will be discussed in the section 5.

Domain Specific Generalization

The performance ratio rises from 94.45% to 96.36% for BERT and from 95.15% to 97.35% for MiniLM when we exclude the domain-specific BIOSSES dataset, which comprises biomedical data. However, it should be noted that the model has not seen any domain-specific data other than that included in the general Wikipedia subset and the STS-Benchmark dataset. Furthermore, the measured speed-up of the forward pass is three times faster. This is because the BERT Tokenizer tends to overdivide scientific words and concepts. The Figure 3 shows an overtokenization of domain-specific language inputs such as Biomedical terms. Although the drop in performance is significant, our observations regarding the benefits of adaptation lead us to hypothesize that the merging procedure with an adaptation phase could be highly beneficial for specialized vocabulary.

5. Conclusions

To improve the semantic representation of texts, we proposed a method for merging vector representations. Based on the assumption that constituent analysis provides semantically relevant information, we merge the token representations of the constituents. This method yields interesting results with BERT and mixed results with MiniLM. In both cases, sequence compression is significant, suggesting the need for further experiments.

We must acknowledge certain limitations to our approach. Token merging is only applied to neighboring tokens in the input sequence. It is difficult to consider the merging of two disjoint tokens within the framework of language models. However, even if the constituent analysis methods selected here do not account for them, some constituents are discontinuous (Coavoux, 2020). Selecting a linguistic criterion for token merging results in heavy text preprocessing, which reduces the performance gains due to the reduced sequence size. Using faster parsers could greatly improve our method.

The experiments conducted here consider segmentation based on noun, verb, and prepositional phrases. Other types of segmentation may also be of interest and require testing. Indeed, constituent parsing is primarily motivated by English and lacks literature in other languages. The proposed method should be language-agnostic, as it relies solely on merging representations based on external linguistic criteria. However, the experiments were conducted only on English corpora and could benefit from being extended to other languages.

The results obtained with MiniLM raise questions about our pre-training procedure and generalization of performances across models. A corpus containing more tokens or more similar to the model’s pre-training corpus would be interesting. The nature of the corpora can influence the results obtained. It would therefore be relevant to construct a corpus similar to the SynPar corpus from the *STSBenchmark* data to ensure corpus similarity.

To ensure the generalizability of this method across models and architectures, further experiments should be conducted.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgments

This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011014970R1 made by GENCI.).

6. Bibliographical References

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster. *ICLR*, abs/2210.09461, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Cao, K. and Clark, S. Factorising AMR generation through syntax. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2157–2163, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1223. URL https://aclanthology.org/N19-1223/.

Carlin, M., Thomas, S., Jansen, A., and Hermansky, H. Rapid evaluation of speech representations for spoken term discovery. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

Chomsky, N. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956. doi: 10.1109/TIT.1956.1056813.

Coavoux, M. Qu'apporte BERT à l'analyse syntaxique en constituants discontinus ? une suite de tests pour évaluer les prédictions de structures syntaxiques discontinues en anglais (what does BERT contribute to discontinuous constituency parsing ? a test suite to evaluate discontinuous constituency structure predictions in English). In Benzitoun, C., Braud, C., Huber, L., Langlois, D., Ouni, S., Pogodalla, S., and Schneider, S. (eds.), *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pp. 189–196, Nancy, France, 6 2020. ATALA et AFCP. URL https://aclanthology.org/2020.jeptalnrecital-taln.17/.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OUIFPHEgJU.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dong, K., Sun, A., Kim, J.-J., and Li, X. Syntactic multi-view learning for open information extraction. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4072–4083, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.272. URL https://aclanthology.org/2022.emnlp-main.272/.

Gee, L., Rigutini, L., Ernandes, M., and Zugarini, A. Multi-word tokenization for sequence compression. In Wang, M. and Zitouni, I. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 612–621, Singapore, December

2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-industry.58. URL https://aclanthology.org/2023.emnlp-industry.58/.

Huang, H., Zhu, D., Wu, B., Zeng, Y., Wang, Y., Min, Q., and Zhou, X. Over-tokenized transformer: Vocabulary is generally worth scaling. *CoRR*, abs/2501.16975, 2025. doi: 10.48550/ARXIV.2501.16975. URL https://doi.org/10.48550/arXiv.2501.16975.

Huang, P., Zhao, X., Hu, M., Tan, Z., and Xiao, W. T 2 -NER: A two-stage span-based framework for unified named entity recognition with templates. *Transactions of the Association for Computational Linguistics*, 11:1265–1282, 2023. doi: 10.1162/tacl_a_00602. URL https://aclanthology.org/2023.tacl-1.72/.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Kumar, D. and Thawani, A. BPE beyond word boundary: How NOT to use multi word expressions in neural machine translation. In Tafreshi, S., Sedoc, J., Rogers, A., Drozd, A., Rumshisky, A., and Akula, A. (eds.), *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pp. 172–179, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.insights-1.24. URL https://aclanthology.org/2022.insights-1.24/.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y. (eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. MTEB: Massive text embedding benchmark. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In Walker, M., Ji, H., and Stent, A. (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. Stanza: A python natural language processing toolkit for many human languages. In Celikyilmaz, A. and Wen, T.-H. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 101–108, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.14. URL https://aclanthology.org/2020.acl-demos.14/.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Reimers, N., Beyer, P., and Gurevych, I. Task-oriented intrinsic evaluation of semantic textual similarity. In Matsumoto, Y. and Prasad, R. (eds.), *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 87–96, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1009/.

Schartz, T., Peddinti, V., Bach, F., Jansen, A., Hermansky, H., and Dupoux, E. Evaluating speech features with the minimal-pair abx task : Analysis of the classical mfc/plp pipeline. In *INTERSPEECH 2013 : 14th Annual Conference of the*

International Speech Communication Association, 2013.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Shen, Y., Tan, S., Sordoni, A., Reddy, S., and Courville, A. Explicitly modeling syntax in language models with incremental parsing and a dynamic oracle. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1660–1672, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.132. URL https://aclanthology.org/2021.naacl-main.132/.

Spearman, C. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. ISSN 00029556. URL http://www.jstor.org/stable/1412159.

Tao, C., Liu, Q., Dou, L., Muennighoff, N., Wan, Z., Luo, P., Lin, M., and Wong, N. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 114147–114179. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cf5a019ae9c11b4be88213ce3f85d85c-Paper-Conference.pdf.

Tytgat, J., Wisniewski, G., and Betrancourt, A. Évaluation de la similarité textuelle : Entre sémantique et surface dans les représentations neuronales. In Balaguer, M., Bendahman, N., Ho-dac, L.-M., Mauclair, J., G Moreno, J., and Pinquier, J. (eds.), *Actes de la 31ème Conférence sur le Traitement Automatique des Langues Naturelles, volume 1 : articles longs et prises de position*, pp. 85–96, Toulouse, France, 7 2024. ATALA and AFPC. URL https://aclanthology.org/2024.jeptalnrecital-taln.6.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xu, Z., Guo, D., Tang, D., Su, Q., Shou, L., Gong, M., Zhong, W., Quan, X., Jiang, D., and Duan, N. Syntax-enhanced pre-trained model. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5412–5422, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.420. URL https://aclanthology.org/2021.acl-long.420/.