The Impact of Word Splitting on the Semantic Content of Contextualized Word Representations

Aina Garí Soler[1] **Matthieu Labeau**[1] **Chloé Clavel**[2]
[1]LTCI, Télécom-Paris, Institut Polytechnique de Paris, France [2]INRIA, Paris, France
{aina.garisoler,matthieu.labeau}@telecom-paris.fr
chloe.clavel@inria.fr

Abstract

When deriving contextualized word representations from language models, a decision needs to be made on how to obtain one for out-of-vocabulary (OOV) words that are segmented into subwords. What is the best way to represent these words with a single vector, and are these representations of worse quality than those of in-vocabulary words? We carry out an intrinsic evaluation of embeddings from different models on semantic similarity tasks involving OOV words. Our analysis reveals, among other interesting findings, that the quality of representations of words that are split is often, but not always, worse than that of the embeddings of known words. Their similarity values, however, must be interpreted with caution.

1 Introduction

With the appearance of pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), there has been an interest in extracting, analyzing, and using contextualized word representations derived from these models, for example, to understand how well they represent the meaning of words (Garí Soler et al., 2019) or to predict diachronic semantic change (Giulianelli et al., 2020).

Most modern PLMs, however, operate at the subword level—they rely on a subword tokenization algorithm to represent their input, like WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016) or Byte Pair Encoding (BPE) (Sennrich et al., 2016). This way of representing words has advantages: With a fixed, reasonably-sized vocabulary (OOV), models can account for out-of-vocabulary words by splitting them into smaller units. When it comes to obtaining representations for words, a subword vocabulary implies that not all words are created equally. Words that have to be split (‘‘split-words’’) need a special treatment, different from words that have a dedicated embedding (‘‘full-words’’).

There are reasons to believe that the semantics of split-words is more poorly represented than that of full-words. First, it is generally assumed that longer tokens tend to contain more semantic information about a word (Church, 2020) because they are more discriminative. The subword representations making up split-words must be able to encode the semantics of all words they can be part of. It has also been noted that tokenization algorithms tend to split words in a way that disregards language morphology (Hofmann et al., 2021), and some of them favor splittings with more subword units than would be necessary (Church, 2020). In fact, a more morphology-aware segmentation seems to correlate with better results on downstream NLP tasks (Bostrom and Durrett, 2020).

In this study, we investigate the impact that word splitting (and how we decide to deal with it) has on the quality of contextualized word representations. We rely on the task of lexical semantic similarity estimation, which has traditionally been used as a way of intrinsically evaluating different types of word representations (Landauer and Dumais, 1997; Hill et al., 2015). We set out to answer two main questions:

- What is the best strategy to combine contextualized subword representations into a contextualized word-level representation?
- (Given a good strategy), how does the quality of split-word representations compare to that of full-word representations?

We design experiments that allow us to answer these and related questions for BERT and other English models. Contrary to previous work

Transactions of the Association for Computational Linguistics, vol. 12, pp. 299–320, 2024. https://doi.org/10.1162/tacl_a_00647
Action Editor: Roberto Navigli. Submission batch: 11/2023; Revision batch: 1/2024; Published 4/2024.

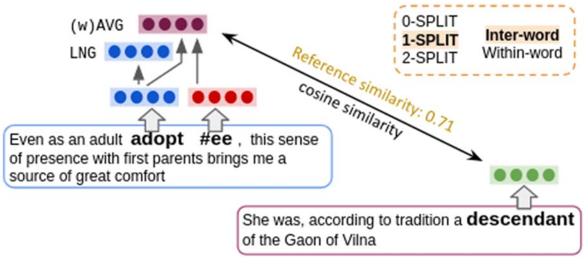
(w)AVG
LNG
0-SPLIT
1-SPLIT
2-SPLIT
Inter-word
Within-word
Reference similarity: 0.71
cosine similarity
Even as an adult adopt #ee , this sense of presence with first parents brings me a source of great comfort
She was, according to tradition a descendant of the Gaon of Vilna

Figure 1: Example of one of our settings where we calculate the cosine similarity between the representations of an OOV word and a known word. We test different ways of creating one embedding for an OOV word (§4), such as AVG and LNG, on two similarity tasks (§3).

where the quality of the lexicosemantic knowledge encoded in word representations is analyzed regardless of the words' tokenization (Wiedemann et al., 2019; Bommasani et al., 2020; Vulić et al., 2020), we analyze the quality of the similarity estimations for split- and full-words separately, and do so in an inter-word and a within-word[1] similarity setting. See Figure 1 for an example of an experimental setting we consider. We uncover several interesting, and sometimes unexpected, tendencies: for example, that when it comes to polysemous nouns, OOV words are better represented than in-vocabulary ones, and that similarity values between two split-words are generally higher than between two full-words. We additionally contribute a new WordNet-based word similarity dataset with a large representation of split-words.[2]

2 Background

Subword tokenization algorithms were first proposed by Schuster and Nakajima (2012) and became widespread after the adaptation of BPE to word segmentation (Gage, 1994; Sennrich et al., 2016). Given a specified vocabulary size, these algorithms create a vocabulary such that the most frequent character sequences in a given corpus can be represented with a single token. Unambiguous detokenization (i.e., recovering the original sequence) can be ensured in different ways. For example, when BERT's tokenizer splits an unknown word into multiple subwords, all but the first are marked with ''##''—we will refer to these as ''sub-tokens'' (as opposed to ''full-tokens'' which do not start with ''##'').

Subword tokenization presented itself as a good compromise between character-level and word-level models, balancing the trade-off between vocabulary size and sequence length. Character-based representations are generally better than subword-based models at morphology, part-of-speech (PoS) tagging, and at handling noisy input and out-of-domain words; but the latter are generally better at handling semantics and syntax (Keren et al., 2022; Durrani et al., 2019; Li et al., 2021a). Because of these advantages, most modern PLMs rely on subword tokenization: BERT uses Wordpiece; RoBERTa, XLM (Conneau and Lample, 2019), GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) use BPE or some variant; T5 (Raffel et al., 2020) relies on SentencePiece (Kudo and Richardson, 2018).

Several studies have pointed out that splitting words may be detrimental for certain tasks, especially if segmentation is not done in a linguistically correct way. Bostrom and Durrett (2020) compare two subword tokenization algorithms, BPE and unigramLM (Kudo, 2018), and find that the latter, which aligns better with morphology, also yields better results on question answering, textual entailment, and named entity recognition. Work on machine translation has shown benefits from using linguistically informed tokenization (Huck et al., 2017; Mager et al., 2022) as well as algorithms that favor segmentation into fewer tokens (Gallé, 2019). In fact, Rust et al. (2021) note that multilingual BERT's (mBERT) tokenizer segments much more in some languages than others, and they demonstrate that a dedicated monolingual tokenizer plays a crucial role in mBERT's performance on numerous NLP tasks. Similarly, Mutuvi et al. (2022) show that increased fertility (i.e., the average number of tokens generated for every word) and number of split-words correlate negatively with mBERT's performance on epidemiologic watch through multilingual event extraction. However, the effect that (over)splitting words—or doing so disregarding their morphology—has on similarity remains unclear.

Nayak et al. (2020) explore a similar question to ours using the BERT model, but compare the similarity between a word representation and their sub-token counterpart (e.g., *night* with *##night*). We argue, however, that even if they represent

[1] Following Liu et al. (2020)'s terminology.

[2] https://github.com/ainagari/splitsim.

the same string, sub-tokens and full-tokens have different distributions and the similarity between them is not necessarily expected to be high.[3] Their experiments additionally involve a modification of the tokenizer. We instead compare representations of whole words using the models' default tokenization, and we work with representations of words extracted from sentential contexts and not in isolation.

Multiple approaches have been proposed to improve on the weak aspects of vanilla subword tokenization, such as the representation of rare, out-of-domain, or misspelled words (Schick and Schütze, 2020b; Hong et al., 2021; Benamar et al., 2022), and its concurrence with morphological structure (Hofmann et al., 2021). Hofmann et al. (2022) devise FLOTA, a simple segmentation method that can be used with pre-trained models without the need for re-training a new model or tokenizer. It consists in segmenting words prioritizing the longest substrings available, omitting part of the word in some cases. FLOTA was shown to match the actual morphological segmentation of words more closely than the default BERT, GPT-2, and XLNet tokenizers, and yielded an improved performance on a topic-based text classification task. El Boukkouri et al. (2020) propose CharacterBERT, a modified BERT model with a character-level CNN intended for building representations for complex tokens. The model improves BERT's performance on several tasks on the medical domain. We test the FLOTA method and the CharacterBERT model in our experiments to investigate their advantages when it comes to lexical semantic similarity.

The split-words in our study are existing words—we do not include misspelled terms—with a generally low frequency. There has been extensive work in NLP focused on improving representations of rare words, which are often involved in lower-quality predictions than those of more frequent words (Luong et al., 2013; Bojanowski et al., 2017; Herbelot and Baroni, 2017; Prokhorov et al., 2019), also in BERT (Schick and Schütze, 2020b). Our goal is not to study the quality of rare word representations per se, but rather the effect of the splitting procedure on the quality of similarity estimates. Given the strong link between splitting and frequency, we also include an analysis controlling for this factor.

3 Similarity Tasks and Data

We evaluate the representations' lexical semantic content on two similarity tasks. In this section we describe the creation of an inter-word similarity dataset (§3.1) as well as the dataset used in our within-word similarity experiments (§3.2).

3.1 Inter-word: The SPLIT-SIM Dataset

We want a dataset annotated with inter-word similarities which allows us to compare similarity estimation quality in three different scenarios: when no word in a pair is split (0-SPLIT), when only one word in a pair is split (1-SPLIT), and when the two words are split (2-SPLIT). We refer to these situations, defined according to a given tokenizer, as ''split-types''.

Factors Affecting Similarity It is well known that, even in out-of-context (OOC) settings (i.e., when comparing word types and not word instances), BERT similarity predictions are more reliable when obtained from a context instead of in isolation (Vulić et al., 2020). However, as shown in Garí Soler and Apidianaki (2021), representations reflect the sense distribution found in the contexts used as well as the words' degree of polysemy. Additionally, it is desirable to take PoS into account, because the quality of similarities obtained with BERT varies across PoS (Garí Soler et al., 2022). To control for all these factors affecting similarity estimates, we conduct separate analyses for words of different nature: monosemous nouns (M-N), monosemous verbs (M-V), polysemous nouns (P-N), and polysemous verbs (P-V). The number of senses of a word with a specific PoS is determined with WordNet (Fellbaum, 1998).

Limitations of Existing Datasets Existing context-dependent (i.e., not OOC) inter-word similarity datasets, like CoSimLex (Armendariz et al., 2020) and Stanford Contextual Word Similarity (SCWS) (Huang et al., 2012) do not have a large enough representation of split-words: With BERT's default tokenization, 97% and 85% of inter-word pairs, respectively, are of type 0-SPLIT. OOC word similarity datasets do not meet our criteria either. In Simlex-999 (Hill et al., 2015) and

[3] For example, in *hitchhiking* (tokenized {*hitch*, *##hi*, *##king*}, *##king* is not semantically related to the word *king*.

WS353 (Agirre et al., 2009), 96% and 95% pairs are 0-SPLIT. CARD-660 (Pilehvar et al., 2018), which specifically targets rare words, has a better distribution of split-types, but it contains a large number of multi-word expressions (MWEs) and lacks PoS information. The Rare Word (RW) dataset (Luong et al., 2013) is also specialized on rare words and has a larger coverage of 1- and 2-SPLIT pairs, but we do not use it because of its low inter-annotator agreement and problems with annotation consistency described in Pilehvar et al. (2018).

Therefore, and since it is more convenient to obtain similarity annotations out-of- rather than in-context, we create a dataset of OOC word similarity, SPLIT-SIM. It consists of four separate subsets, one for each type of word. Each subset has a balanced representation of split-types.

Word Selection and Sentence Extraction We use WordNet to create SPLIT-SIM. We first identify all words in WordNet which are not MWEs, numbers or proper nouns, and which are at least two characters long. After this filtering, we find 28,563 monosemous nouns, 12,903 polysemous nouns, 3,888 monosemous verbs, and 4,518 polysemous verbs.

We search for sentences containing these words in the c4 corpus (Raffel et al., 2020), from which we will derive contextualized word representations. We postag sentences using `nltk` (Bird et al., 2009).[4] Importantly, we only select sentences that contain the lemma form of a word with the correct PoS. This ensures that a word will be tokenized in the same way (and belong to the same split-type) in all its contexts, and avoids BERT's word form bias (Laicher et al., 2021). We only keep words for which we could find at least ten sentences that are between 5 and 50 words long. If we found more, we randomly select 10 sentences among the first 100 occurrences found.

Pair Creation We rely on WUP (Wu and Palmer, 1994), a Wordnet-based similarity measure, as our reference similarity value. WUP similarity takes into account the depth (the path length to the root node) of the two senses to be compared (s_1 and s_2), as well as of their ''least common subsumer'' (LCS). In general, the deeper LCS is, the higher the similarity between s_1 and s_2.[5]

Dataset	PoS	ρ	# pairs
Simlex-999	n	0.55	666
	v	0.39	162
WS353	n	0.64	201
	v	0.10	29
CARD-660	n	0.64	170
	v	0.50	20
RW	n	0.24	910
	v	0.25	681

Table 1: Spearman's ρ between WUP similarity and human judgments from existing word similarity datasets.

WUP similarities are only available for nouns and verbs. It is important to note that similarities for the two PoS follow slightly different distributions, which is another reason for keeping them separate. We choose WUP over other WordNet-based similarity measures like LCH (Leacock et al., 1998) and path similarity because it conveniently ranges from 0 to 1 and its distribution aligns with the intuition that most randomly obtained pairs would have a low semantic similarity.[6] WUP is not as good as human judgments, but it correlates reasonably well with them (Yang et al., 2019a). Table 1 shows the measure's correlation with manual similarity judgments by PoS. We consider it to be a good enough approximation for our purposes of comparing performance across split-types and representation strategies. For an alternative non-Wordnet-based similarity metric to compare to WUP, we also use the similarity of FastText embeddings (Bojanowski et al., 2017) as a control.

We exhaustively pair all words in each subset and calculate their WUP similarity. We select a portion of all pairs ensuring that the full spectrum of similarity values is represented: For each split-type, we randomly sample the same number of word pairs in each 0.2-sized similarity score

[4] `nltk` offers a good speed/accuracy trade-off compared with SpaCy, Flair (Akbik et al., 2019), stanza (Qi et al., 2020) and the RDRPOSTagger (Nguyen et al., 2014). The agreement between the `nltk` and SpaCy tags for the target words in our final set of selected sentences is of 89.8%.

[5] Since WUP is a sense similarity measure, we define the similarity of two polysemous words to be the highest similarity found between all possible pairings of their senses.

[6] We observed the distribution of similarity values of the three measures on a random sample of 2,000 lemmas. Similarities are calculated using `nltk`.

			M-N	M-V	P-N	P-V
full	BERT	0-split	22,500	850	5,000	5,000
		1-split	22,500	850	5,000	5,000
		2-split	22,500	850	5,000	5,000
	XLNet	0-split	12,166	644	3,642	5,610
		1-split	25,490	1,033	6,009	6,006
		2-split	29,844	873	5,349	3,384
		Total	67,500	2,550	15,000	15,000
balanced	BERT	0-split	7,387	122	572	240
		1-split	3,873	119	973	687
		2-split	1,915	146	1,553	1,776
	XLNet	0-split	2,491	74	317	563
		1-split	5,992	165	1,149	1,270
		2-split	4,692	148	1,632	870
		Total	13,175	387	3,098	2,703

Table 2: Composition of the split-sim dataset (full and balanced versions) according to two different tokenizers.

Word pairs		**Split-type**	wup
{accordion}	{guitar}	0-split	0.80
{tom, ##fo, ##ole, ##ry}	{loaf, ##ing}	2-split	0.63
{ethanol}	{fuel}	0-split	0.46
{ash, ##tray}	{weather}	1-split	0.24

Table 3: Example word pairs from split-sim (m-n subset) with their BERT tokenization.

interval. Due to data availability this number is different for each subset. For the creation of the dataset, the split-type is determined using BERT's default tokenization. Table 2 contains statistics on the full dataset composition. Example pairs from the dataset can be found in Table 3.

Controlling for Frequency In our experiments we also want to control for frequency, since split-words tend to be more rare than full-words. We calculate the frequencies of words in split-sim with the `wordfreq` Python package (Speer, 2022) and report them in Table 4. Frequencies are low overall, especially those of monosemous split-words. To mitigate the potential effect of frequency differences, we find the narrowest possible frequency range that is still represented with enough word pairs in every split-type. We determine this range to be [2.25, 3.75). We create a smaller version of split-sim, which we call ‘‘balanced’’, with pairs that include only words within this frequency interval. Another aspect to take into account is that of the difference in frequencies of words in a pair, what we call Δf. Δf is highest in 1-split pairs (up to 2.19 in m-v compared to 0.67 in the corresponding 0-split), but it is much lower overall in the balanced dataset because of the narrower frequency range.

		M-N	M-V	P-N	P-V
full	0-split	3.75	3.99	4.09	4.30
	1-split	2.66	2.93	3.15	3.27
	2-split	1.54	1.81	2.18	2.25
balanced	0-split	3.35	3.38	3.43	3.51
	1-split	3.04	3.09	3.14	3.19
	2-split	2.72	2.81	2.84	2.90

Table 4: Average frequencies in each split-sim subset (BERT tokenization). Values are the base-10 logarithm of the number of times a word appears per billion words. For reference, the frequencies of *can*, *dog*, *oatmeal* and *myxomatosis* are 6.46, 5.10, 3.37, and 1.61.

3.2 Within-word

Similarly to the inter-word setting, for within-word similarity we want to distinguish between 0-, 1- and 2-split pairs. An important factor that can influence within-word similarity estimations is whether pairs compare the same word form (same) or different morphological forms of the word (diff). 1-split pairs are all necessarily of type diff,[7] but 0- and 2-split pairs can be of either type (e.g., {*carry*} vs {*carries*}; {*multi*, *##ply*} vs {*multi*, *##ply*, *##ing*}).

We choose the Word-in-Context (WiC) dataset (Pilehvar and Camacho-Collados, 2019) for its convenient representation of all split-types. WiC contains pairs of word instances that have the same (T) or a different (F) meaning. We use the training and development sets, whose labels (which are taken as a reference) are publicly available. They consist of a total of 6,066 pairs that we rearrange for our purposes. We use as training data all 0-split pairs found in the original training set. For evaluation we use the 0-split pairs

[7] Except for XLNet, which is a cased model.

		Training	Evaluation		
		0-SPLIT	0-SPLIT	1-SPLIT	2-SPLIT
BERT	**All**	5,104	479	117	366
	SAME	3,388	312	0	274
	DIFF	1,716	167	117	92
	T	2,464	228	72	269
	F	2,640	251	45	97
	Lemmas	1,043	445	102	288
XLNet	**All**	4,648	415	502	501
	SAME	3,144	292	142	396
	DIFF	1,504	123	360	105
	T	2,272	203	222	336
	F	2,376	212	280	165
	Lemmas	944	387	291	351

Table 5: WiC statistics: Number of word pairs of different types and number of unique lemmas with different tokenizers.

in the original development set, and all 1-SPLIT and 2-SPLIT pairs found in both sets. Table 5 contains details about the composition of the dataset, such as the proportion of T and F labels. Note that, again, numbers differ depending on the tokenizer used (BERT's or XLNet's).

WiC is smaller than SPLIT-SIM and offers a less controlled, but more realistic, environment. For example, 2-SPLIT pairs involve words with low frequency and few senses, which results in an overrepresentation of T pairs in this class. We did not use other within-word similarity datasets such as Usim (Erk et al., 2009, 2013) or DWUG (Schlechtweg et al., 2021), because they contain a small number of 1- and 2-SPLIT pairs (91 and 4 in Usim), or these involve very few distinct lemmas (14 and 12 in DWUG).

4 Experimental Setup

4.1 Models

We run all our experiments with representations extracted from the BERT (base, uncased) model in the `transformers` library (Wolf et al., 2020) and the `general` CharacterBERT model (hereafter CBERT).[8] The two are trained on a comparable amount of tokens (3.3B and 3.4B, respectively) which include English Wikipedia. BERT is also trained on BookCorpus (Zhu et al., 2015), and CBERT on OpenWebText (Gokaslan and Cohen, 2019). For comparison, we also include ELECTRA base (Clark et al., 2020) and XLNet (base, cased)[9] (Yang et al., 2019b) in our analysis. ELECTRA is trained on the same data as BERT and uses exactly the same architecture, tokenizer, and vocabulary (30,522 tokens), but is trained with a more efficient discriminative pre-training approach. XLNet relies on the SentencePiece implementation of UnigramLM and has a 32,000 token vocabulary. It is a Transformer-based model pre-trained on 32.89B tokens with the task of Permutation Language Modeling. We choose these models because they are newer and better than BERT (e.g., on GLUE (Wang et al., 2018) among other benchmarks) and because of their wide use. XLNet allows us to investigate the effect of word splitting in models relying on different tokenizers. We experiment with all layers of the models. In inter-word experiments, a word representation is obtained by averaging the contextualized word representations from each of the 10 sentences.

4.2 Input Treatment

Here we describe the different ways in which input data is processed before feeding it to the models.

Tokenization We use the model's default tokenizations. We additionally experiment with the FLOTA tokenizer (Hofmann et al., 2022) used in combination with BERT. FLOTA has a hyperparameter controlling the number of iterations, $k \in \mathbb{N}$. With lower k, portions of words are more likely to be omitted. We set k to 3 as it obtained the best results on text classification (Hofmann et al., 2022).

Lemmatization In the WiC dataset, the word instances to be compared may have different surface forms. One way of restricting the influence of word form on BERT representations is through lemmatization (Laicher et al., 2021). We replace the target word instance with its lemma before extracting its representation. We refer to this setting as `LM`. This procedure is not relevant for

[8] `https://github.com/helboukkouri/character-bert`.

[9] The cased and uncased versions of a word may be split differently. To avoid inconsistencies in the definition of split-types in SPLIT-SIM, target words are presented in lower case exclusively.

	BERT			BERT-FLOTA			CBERT	ELECTRA			XLNet		
	AVG	WAVG	LNG	AVG	WAVG	LNG	–	AVG	WAVG	LNG	AVG	WAVG	LNG
M-N	38_{6}	38_{6}	31_{8}	35_{5}	35_{6}	30_{8}	40_{10}	39_{5}	40_{5}	35_{5}	41_{10}	$\mathbf{42}_{4}$	$\mathbf{42}_{4}$
M-V	33_{11}	33_{11}	31_{12}	27_{12}	28_{12}	25_{12}	31_{3}	34_{5}	35_{3}	28_{5}	36_{4}	$\mathbf{37}_{4}$	$\mathbf{37}_{4}$
P-N	33_{10}	34_{10}	29_{12}	28_{10}	28_{10}	26_{12}	29_{10}	34_{8}	35_{6}	32_{7}	35_{10}	36_{10}	$\mathbf{37}_{5}$
P-V	30_{10}	30_{12}	28_{12}	24_{12}	24_{12}	21_{12}	25_{10}	27_{8}	28_{8}	26_{7}	29_{7}	31_{6}	$\mathbf{33}_{4}$

Table 6: Spearman's ρ ($\times$ 100) obtained on SPLIT-SIM with different representation types and strategies. Subscripts denote the best layer. The best result on each subset is boldfaced.

SPLIT-SIM, where all instances are already in lemma form.

4.3 Split-words Representation Strategy

We compare different strategies for pooling a single word embedding from the representations of a split-word's multiple subwords.

Average (AVG) The embeddings of every subword forming a word are averaged to obtain a word representation. This is the most commonly used strategy when representing split-words (Wiedemann et al., 2019; Garí Soler et al., 2019; Liu et al., 2020; Montariol and Allauzen, 2021, inter alia). Bommasani et al. (2020) tested max, min, and mean pooling as well as using the representation of the last token. We only use mean pooling (AVG) from their work because they found it to work best for OOC word similarity.

Weighted Average (WAVG) A word is represented with a weighted average of all its subword representations. Weights are assigned according to word length. For example, a subword that makes up 70% of a word's characters is weighted with 0.7.

Longest (LNG) Only the representation of the longest subword is used. This approach, as WAVG, accounts for the intuition that longer pieces carry more information about the meaning of a word.

4.4 Prediction and Evaluation

The similarity between two words or word instances is calculated as the cosine similarity between their representations. For experiments on SPLIT-SIM, the evaluation metric is Spearman's ρ. For within-word experiments, we train a logistic regression classifier that uses the cosine between two word instance representations as its only feature. We evaluate the classifier based on its accuracy.

5 Results and Analysis

In this section we analyze the results obtained on the SPLIT-SIM (§5.1) and WiC (§5.2) datasets.

5.1 Inter-word

We start with a look at the results of each method on each SPLIT-SIM subset as a whole. The rest of this section is organized around the main questions we aim to answer.

Table 6 presents the correlations obtained by different representation types and strategies on the full dataset. We report the highest correlation found across all layers. The best model on all subsets is clearly XLNet with the LNG or WAVG strategies. ELECTRA (with WAVG) is the second best one on most subsets. Correlations obtained against FastText cosine similarities reflect, with few exceptions, the same tendencies observed in this section (results are presented in Appendix A).

5.1.1 What Is the Best Strategy to Represent Split-words?

Table 7 shows the Spearman's correlations obtained by different pooling methods on the three split-types. The best layer is selected separately for each split-type, model, and strategy. We can see that the best strategy for each model tends to be stable across datasets. AVG is the preferred strategy overall, followed by WAVG, which, in ELECTRA and XLNet, performs almost on par with AVG. Using the longest subword (LNG) results in a considerably lower performance across models and data subsets, presumably because some important information is excluded from the representation. CBERT obtains good results (comparable or better than BERT) on monosemous nouns (M-N), but on other kinds of words it generally lags behind.

		BERT			BERT-FLOTA			CBERT	ELECTRA			XLNet		
		AVG	WAVG	LNG	AVG	WAVG	LNG	–	AVG	WAVG	LNG	AVG	WAVG	LNG
	0-s	49			49			48	48			51		
M-N	1-s	**41***	38*	28*	**35***	33*	26*	41*	**42***	40*	35*	**46***	**46***	44*
	2-s	**43***	40*	26*	**34***	32*	23*	47	**45***	43*	31*	**46***	45*	39*
	0-s	43			43			39	42			50		
M-V	1-s	**33***	**33***	26*	**23***	**23***	19*	28*	**36**	**36**	26*	**34***	**34***	32*
	2-s	**41**	40	32*	**25***	**25***	23*	35	36	**38**	28*	**39***	37*	35*
	0-s	38			38			31	32			38		
P-N	1-s	**38**	35	28*	**32***	30*	24*	31	**38***	37*	34	**39**	38	37
	2-s	**41***	37	25*	**29***	27*	20*	43*	**45***	44*	36*	**45***	43*	39
	0-s	37			37			31	34			37		
P-V	1-s	**34***	33*	25*	**26***	23*	18*	25*	**30***	**30***	27*	**35**	33*	32*
	2-s	**33***	31*	24*	**16***	15*	14*	31	31*	**32**	26*	**34**	33*	31*

Table 7: Spearman's ρ ($\times$ 100) on SPLIT-SIM (full). The best result by subset, split-type, and model is boldfaced. The best overall result in every subset and split-type is underlined. * indicates that a 1- or 2-SPLIT correlation coefficient is significantly different ($\alpha < 0.05$) from the corresponding 0-SPLIT result (Sheskin, 2003).

		AVG		WAVG		LNG	
		COM	INCM	COM	INCM	COM	INCM
M-N	1-s	**36**	30	**33**	30	26	**30**
	2-s	31	**41**	29	**40**	20	**28**
M-V	1-s	22	**25**	**23**	22	**20**	03
	2-s	23	**32**	24	**31**	22	**27**
P-N	1-s	**33**	22	**30**	29	**24**	21
	2-s	**30**	24	**28**	24	**21**	17
P-V	1-s	**26**	16	**24**	09	**19**	−02
	2-s	**17**	09	**17**	07	**15**	07

Table 8: Results obtained with FLOTA tokenization on pairs where words were fully preserved (COM) and where at least one word had a portion omitted (INCM).

FLOTA Performance The use of the FLOTA tokenizer systematically decreases BERT's performance. We believe there are two main reasons behind this outcome: First, that similarly to LNG, FLOTA sometimes[10] omits parts of words. We investigate this by comparing its performance on pairs where both words were left complete (COM) to that on pairs where some word is incomplete (INCM). We present results in Table 8. We observe that, indeed, in most cases, performance is worse when parts of words are omitted. However, this is not the only factor at play, since the performance on COM is still lower than when using BERT's default tokenizer. The second reason, we believe, is that FLOTA tokenization differs from the tokenization used for BERT's pretraining. FLOTA was originally evaluated on a *supervised* text classification task (Hofmann et al., 2022), while we do not fine-tune the model for similarity estimation with the new tokenization. Additionally, classification was done relying on a sequence-level token representation (e.g., [CLS] in BERT). It is possible that FLOTA tokenization provides an advantage when considering full sequences which does not translate to an improvement in the similarity between individual word token representations. Given its poor results compared with BERT, in what follows, we omit FLOTA from our discussion.

5.1.2 Is Performance on Pairs Involving Split-words Worse Than on 0-SPLIT?

In Table 7 we can see that, as expected, in most subsets (M-N, M-V, and P-V), performance is worse in pairs involving split-words. This is, however, not true of polysemous nouns (P-N), where similarities obtained with all models are of better or comparable quality on 1- and 2-SPLIT pairs. With CBERT, performance on 2-SPLIT pairs is never significantly lower than on 0-SPLIT pairs.

Lower Correlation of Polysemous Words Correlations obtained on polysemous words are

[10] With FLOTA, 9.8% to 20.8% of 1- and 2-SPLIT pairs (depending on the dataset) have at least one incomplete word.

overall lower than on monosemous words, particularly so in the 0-SPLIT case. Worse performance on polysemous words can be expected for two main reasons. First, WUP between polysemous words is determined as the maximum similarity attested for all their sense pairings, while cosine similarity takes into account all the contexts provided as well as the accumulated lexical knowledge about the word contained in the representation. Second, the specific sense distribution found in the randomly selected contexts may also have an impact on the final results (particularly if, e.g., the relevant sense for the comparison is missing).

1-SPLIT VS 2-SPLIT Another interesting observation is that, in most cases, performance on 1-SPLIT pairs is lower than on 2-SPLIT pairs. We identify two main factors that explain this result. One is the fact that in 1-SPLIT, the words in a pair are represented using different strategies (the plain representation vs {AVG | WAVG | LNG}). In fact, exceptions to this observation concern almost exclusively the LNG pooling strategy. LNG does not involve any arithmetic operation, which makes the representations of the split- and full-word in a 1-SPLIT pair more comparable to each other. Another explanation is the difference in frequency between words (Δf), which tends to be larger in 1-SPLIT than in 0- and 2-SPLIT pairs. We explore this possibility in our frequency analysis below.

In the remaining inter-word experiments, we focus our observations on the better (and simpler) AVG strategy.

5.1.3 Frequency-related Analysis

As explained in Section 3.1, frequency and word-splitting are strongly related. The experiments presented in this section help us understand how the tendencies observed so far are linked to or affected by word frequency.

Controlling for Frequency The lower correlations obtained in 1- and 2-SPLIT pairs in most subsets could simply be due to the lower frequency of split-words, and not necessarily to the fact that they are split. To verify this, we evaluate the models' predictions on word pairs found in the balanced SPLIT-SIM. Results are presented in Table 9. When comparing 0-SPLIT pairs to pairs involving split-words, we observe the same tendencies as in the full version of SPLIT-SIM: For monosemous words and polysemous verbs, word splitting has a negative effect on word representations. There are, however, some differences in the significance of results, particularly in P-V, due in part to the much smaller sample size of this dataset.

		BERT	CBERT	ELECTRA	XLNet
	0-S	**52**	**52**	**53**	**57**
M-N	1-S	47*	49*	49*	53*
	2-S	44*	47*	48*	49*
	0-S	**53**	**54**	**60**	**71**
M-V	1-S	39	32*	31*	46*
	2-S	42	32*	40*	36*
	0-S	39	41	44	47
P-N	1-S	**45**	**46**	**46**	**48**
	2-S	41	40	44	42
	0-S	**46**	**46**	**46**	**48**
P-V	1-S	39	37	44	46
	2-S	39	35*	40	40*

Table 9: Spearman's ρ ($\times$ 100) on SPLIT-SIM (balanced), AVG strategy. The best result by subset and model is boldfaced. * indicates that a 1- or 2-SPLIT correlation coefficient is significantly different from the corresponding 0-SPLIT result (Sheskin, 2003).

It is important to note that split-types are strongly defined and determined by word frequency. In natural conditions (i.e., without controlling for frequency), we expect to encounter the patterns found in Table 7.

The Effect of Δf In Table 9, we can see that, in a dataset with lower and better balanced Δf values, 1-SPLIT pairs are no longer at a disadvantage and obtain results that are most of the time superior to those of 2-SPLIT pairs. We run an additional analysis to study the effect of different Δf. We divide the pairs in each subset and split-type according to whether their Δf is below or above a threshold $t = 0.25$, ensuring that all sets compared have at least 100 pairs. Results, omitted for brevity, show that pairs with lower Δf obtain almost systematically better results than those with higher Δf. This confirms that a disparity in the frequency levels of the words compared also has a negative effect on similarity estimation.

		BERT		CBERT		ELECTRA		XLNet	
		L	H	L	H	L	H	L	H
M-N	0-s	52	52	51	51	**52**	51	**59**	53
	1-s	44	**49**	45	**51**	47	**51**	**54**	47
	2-s	**47**	42	**54**	45	**49**	46	**52**	47
P-N	0-s	36	**43**	38	**40**	39	**40**	40	**42**
	1-s	45	45	**47**	44	45	**48**	**51**	37
	2-s	**43**	42	**52**	40	**50**	45	**48**	43
P-V	0-s	**40**	39	**39**	38	**41**	39	**48**	38
	1-s	**47**	39	39	**42**	**48**	44	**44**	38
	2-s	36	**40**	**41**	38	36	**41**	**41**	39
Without context									
M-N	0-s	37	**43**	44	**46**	45	**49**	**58**	51
	1-s	24	**29**	39	**42**	30	**32**	31	**32**
	2-s	**29**	28	36	36	**32**	27	**32**	29
P-N	0-s	14	**29**	32	**36**	19	**36**	33	**34**
	1-s	25	**27**	**40**	35	25	**32**	**28**	23
	2-s	22	**28**	**35**	34	**26**	25	**23**	21
P-V	0-s	29	**34**	29	**32**	36	**41**	**44**	38
	1-s	**34**	18	**41**	33	**29**	21	23	**25**
	2-s	19	**27**	31	**34**	15	**31**	22	**25**

Table 10: Results on pairs with low (L) and high (H) frequency using 10 (top) and no (bottom) contexts.

The Effect of Frequency on Similarity Estimation To investigate how estimation quality varies with frequency, we divide the data in every subset and split-type into two sets, L (low) and H (high), based on individually determined frequency thresholds. Using different thresholds does not allow us to fairly compare across data subsets and split-types but ensures that both classes (L and H) are always well-represented and balanced. The frequency of a word pair is calculated as the average frequency of the two words in it. To prevent L and H from containing pairs of similar frequency, their thresholds are apart by 0.25. We only include pairs with a Δf of at most 1. M-V is excluded from this analysis because of its small size.

Table 10 (top section) shows results of this analysis. Very often, correlations are higher on the sets of pairs with lower average frequency (L). This is surprising, because, as explained in Section 2, rare words are typically problematic in NLP. Works investigating the representation of rare words in BERT, however, either test it through prompting (Schick and Schütze, 2020b), on ''rarified'' downstream tasks (Schick and Schütze, 2020a), or on word similarity but without providing contexts (Li et al., 2021b). We believe the observed result is due to a combination of multiple factors, both contextual and lexical. First, the contexts used to extract representations provide information about the word's meaning. If we compare results to a setting where words are presented without context (lower part of Table 10), the tendency is indeed softened, but not completely reversed, meaning that context alone does not fully explain this result. Lower frequency words are also more often morphologically complex than higher frequency ones. This is the case in our dataset.[11] In the case of split-words, morphological complexity may be an advantage that helps the model understand word meaning through word splitting. Another factor contributing to this result may be the degree of polysemy. We have seen in Table 7 that similarity estimation tends to be of better quality on monosemous words than on polysemous words. However, a definite explanation of the observed results would require additional analyses which are beyond the scope of this study.

5.1.4 Further Analysis

How Do Results Change Across Layers for Every Split-type? Figure 2 shows the BERT AVG performance on each split-type of every subset across model layers. In M-N, M-V, and P-V we observe that at earlier layers the quality of the similarity estimations involving split-words is lower than that of 0-SPLIT pairs. However, as information advances through the network and the context is processed, their quality improves at a higher rate than that of 0-SPLIT, which remains more stable. This suggests that split-words benefit from the contextualization process taking place in the Transformer layers more than full-words. This makes sense, since sub-tokens are highly ambiguous (i.e., they can be part of multiple words), so more context processing is needed for the model to represent their meaning well. In a similar vein, the initial advantage of 0-SPLIT pairs is more pronounced in monosemous words, which is expected

[11] We verify this with the MorphoLEX (Sánchez-Gutiérrez et al., 2018) and LADEC (Gagné et al., 2019) databases.

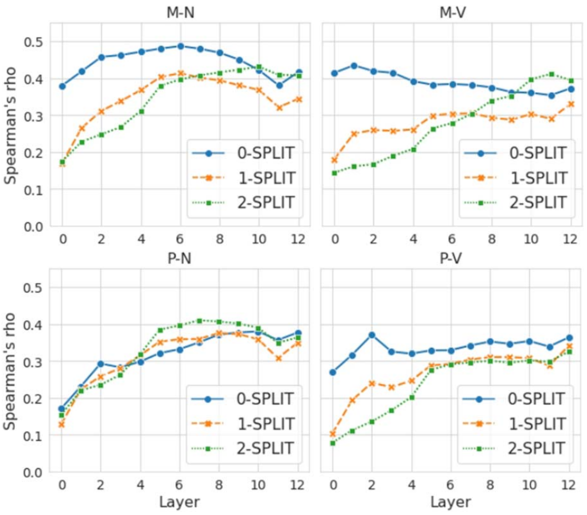

Figure 2: BERT AVG results by layer and split-type on every SPLIT-SIM subset.

		M-N		P-N		P-V	
		1-s	2-s	1-s	2-s	1-s	2-s
BERT	COR	**47**	35	**39**	**52**	34	**48**
	INC	44	**40**	36	38	**35**	32
CBERT	COR	41	38	27	36	**28**	**54**
	INC	**43**	**43**	**31**	**41**	26	30
ELECTRA	COR	**46**	**51**	**41**	**57**	28	**50**
	INC	44	43	37	41	**31**	30
XLNET	COR	**52**	**58**	**40**	**51**	**42**	**44**
	INC	47	43	38	42	35	33

Table 11: Spearman's ρ ($\times$ 100) on pairs with an incorrectly segmented word (INC) and pairs where the root(s) of both words are preserved (COR).

as context is less crucial for understanding their meaning. In P-N, the situation is different: 0-SPLIT pairs behave in a similar way as 1- and 2-SPLIT pairs from the very first layers. We verify whether this could be due to non-split polysemous nouns in P-N being particularly ambiguous. We obtain their number of senses and we also check how many split-words in WordNet they are part of following BERT's tokenization (e.g., the word ‘‘station’’ is part of {station, ##ery}). These figures, however, are higher in P-V, so this hypothesis is not confirmed.

We also note that performance for the different split-types usually peaks at different layers. This highlights the need to carefully select the layer to use depending on the word's tokenization.

The same tendencies are observed with ELECTRA and XLNet. In CBERT, results are much more stable across layers.

Is a Correct Morphological Segmentation Important for the Representations' Semantic Content? As explained in Section 2, the morphological awareness of a tokenizer has a positive effect on results in NLP tasks. Here we verify whether it is also beneficial for word similarity prediction. We use MorphoLex, a database containing morphological information (e.g., segmentation into roots and affixes) on 70,000 English words. We consider that a split-word in SPLIT-SIM is incorrectly segmented if one or more of the roots of the word have been split (e.g., *saltshaker*: {*salts*, *##hak*, *##er*}).[12] We compare the performance on word pairs involving an incorrectly segmented word (INC) to that of pairs where the root(s) are fully preserved in both words (COR), regardless of whether the tokens containing the root contain other affixes (e.g., {*marina*, *##te*}). Note that MorphoLex does not fully cover the vocabulary in SPLIT-SIM.[13] We exclude M-V from this analysis because of the insufficient amount of known COR pairs (4 in 2-SPLIT following BERT's tokenization). All other comparisons involve at least 149 pairs. Results are presented in Table 11. They confirm that, in subword-based models, when tokenization aligns with morphology, representations are almost always of better quality than when it does not. The results obtained with CBERT, evaluated according to BERT's tokenization, highlight that the same set of INC pairs is not necessarily harder to represent than COR for a model that does not rely on subword tokenization.

Do Similarity Predictions Vary Across Split-types? In Figure 3 we show the histogram of similarities calculated with BERT AVG using the best overall layer (cf. Table 6). We observe that similarity values are found in different, though overlapping, ranges depending on the split-type.

[12] We do not base the definition of an incorrectly segmented word on the preservation of affixes because the segmentation in MorphoLex contains versions of affixes that do not always match the form realized in the word (e.g., sporadically = sporadic + ly).

[13] Its coverage ranges between 38% and 76% of words depending on the subset.

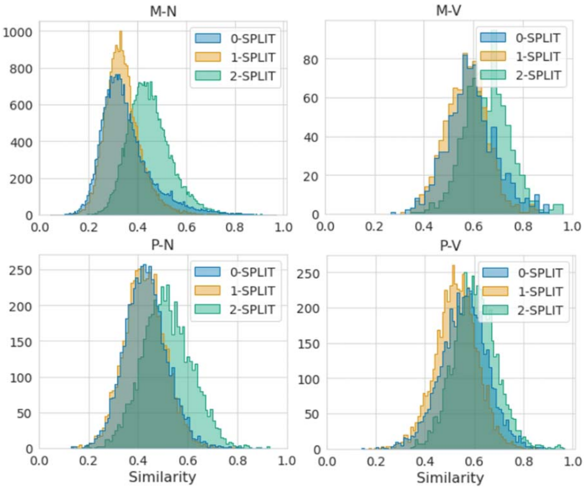

Figure 3: Distribution of predicted similarity values by BERT (AVG) across split-types in SPLIT-SIM.

		BERT		ELECTRA		XLNet	
		−	+	−	+	−	+
M-N	1-s	41	**42**	42	42	**48**	45
	2-s	37	**49**	44	**48**	46	**49**
M-V	1-s	33	**36**	**37**	36	**37**	32
	2-s	35	**51**	31	**44**	35	**42**
P-N	1-s	**38**	37	**40**	34	38	**39**
	2-s	42	42	**46**	44	**49**	44
P-V	1-s	32	**36**	28	**31**	35	35
	2-s	**34**	31	**32**	30	35	**36**

Table 12: Spearman's ρ ($\times$ 100) obtained on SPLIT-SIM pairs tokenized into few (−) or many (+) subwords.

2-SPLIT pairs exhibit a clearly higher average similarity than 0- and 1-SPLIT pairs. Similarities in 1-SPLIT tend to be the lowest, but the difference is smaller. This does not correspond to the distribution of gold WUP similarities, which, due to our data collection process, does not differ across split-types. A possible partial explanation is that sub-token (##) representations are generally closer together because they share distributional properties.[14] The same phenomenon is found in all models tested (ELECTRA, XLNet, and CBERT), but is less pronounced in nouns in XLNET.

This observation has important implications for similarity interpretation, and it discourages the comparison across split-types even when considering words of the same degree of polysemy and PoS. A similarity score that may be considered high for one split-type may be just average for another.

Does the Number of Subwords Have an Impact on the Representations' Semantic Content? We saw in Section 2 that oversplitting words has negative consequences on certain NLP tasks. We investigate the effect that the number of subwords has on similarity predictions. We depart from the hypothesis that the more subwords a word is split into, the worse the performance will be. This is based on the intuition that shorter subwords are not able to encode as much lexical semantic information as longer ones. We count the total number of subwords in each word pair and re-calculate correlations separately on sets of word pairs with few (−) or many (+) subwords. In 1-SPLIT, ''−'' is defined as 3 subwords and in 2-SPLIT, as 5 or less. We make sure that every set contains at least 1,000 pairs. Results are presented in Table 12. Our expectations are only met in about half of the cases, particularly in P-N. Surprisingly, similarity estimations from BERT tend to be more accurate when words are split into a larger number of tokens, even though the tokenization in + is more often morphologically incorrect than in −. Results from other models are mixed.

Since only the first subword in a split-word is a full-token (i.e., does not begin with ## in BERT), one difference between words split into few or many pieces is the ratio of full-tokens to sub-tokens. When using the AVG strategy, on ''−'' split-words, the first subword (a sub-token) has a large impact on the final representation, which is reduced as the number of subwords increases. We investigate whether this difference has something to do with the results obtained with BERT. To do so, we test two more word representation strategies: o1, where we omit the first subword (the full-token) and oL, where we omit the last subword (a sub-token). If mixing the two kinds of subwords (sub-tokens and full-tokens) is detrimental for the final representation, we expect o1 to obtain better results than oL. Results by these two strategies could be affected by the morphological structure of words in SPLIT-SIM (e.g., o1

[14] We indeed find that, in BERT's embedding layer, similarity between random sub-tokens is slightly higher (0.46) than between full-tokens or in mixed pairs (0.44 in both cases).

		M-N		P-N		P-V	
		1-S	2-S	-S	2-S	1-S	2-S
–	o1	**51**	**42**	**33**	32	**30**	36
	oL	42	37	30	**33**	28	**38**
+	o1	**47**	**37**	**42**	26	39	**41**
	oL	43	29	40	**45**	39	37

Table 13: Results obtained with BERT AVG omitting the first (o1) or last (oL) token on simplex SPLIT-SIM pairs tokenized into different amounts of subwords.

could perform better than oL on words with a prefix). To control for this, we only run this analysis on word pairs consisting of two simplexes (according to MorphoLex). We exclude M-V because of the insufficient (< 100) number of pairs available in each class.

Results of this analysis are shown in Table 13. In most cases, particularly in M-N, the o1 strategy, which excludes the only full-token in the word, obtains a better performance than oL. This suggests that, in the BERT model, the first token is less useful when building a representation. This is surprising, because English tends to place disambiguatory cues at the beginning of words (Pimentel et al., 2021), and because the first subword is often the longest one.[15] The intuition that representations of longer tokens contain more semantic information is, thus, not confirmed.

5.2 Within-word

In this section we present the results on the WiC dataset. In Table 14, we report the best accuracy obtained by every model on different split-types. We observe that the best performance is achieved on the full set of 2-SPLIT pairs (ALL). This can be explained by the label distribution in 2-SPLIT, where most pairs are of type T (cf. Table 5). We have seen in Section 5.1 that AVG representations for these pairs have higher similarity values, and we confirm this is the case, too, in the within-word setting (see Figure 4). In fact, in the case of BERT AVG, only 18 out of 97 F 2-SPLIT word pairs were correctly guessed. To have a fairer comparison with 0-SPLIT pairs, where labels are more balanced, we recalculate accuracy on 1- and 2-SPLIT pairs randomly subsampling as many T pairs as the number of available F pairs (BAL). These results are shown in the same Table. From them, we conclude that accuracy on 1- and 2-SPLIT pairs is actually lower than that on 0-SPLIT. This is not true of CBERT, however, which performs equally well across split-types and is the best option for 2-SPLIT pairs. As we can see in Figure 4, the similarities it assigns to 2-SPLIT are in a similar range to 0-SPLIT in this within-word setting.

		0-S	1-S		2-S	
		ALL	ALL	BAL	ALL	BAL
BERT	AVG	70	66	**67**	75	57
	WAVG		65	63	75	58
	LNG		65	62	74	**60**
FLOTA	AVG	69	60	**62**	74	**60**
	WAVG		60	58	75	59
	LNG		60	57	73	**60**
CBERT	–	67	57	67	66	66
ELECTRA	AVG	71	62	**62**	76	58
	WAVG		62	59	76	61
	LNG		57	59	75	**65**
XLNET	AVG	62	61	61	68	**58**
	WAVG		62	**62**	69	**58**
	LNG		62	**62**	68	57

Table 14: Accuracy obtained on WiC on the full subsets (ALL) and balancing T/F labels in 1- and 2-SPLIT (BAL). The best result per model and split-type in BAL subsets is boldfaced.

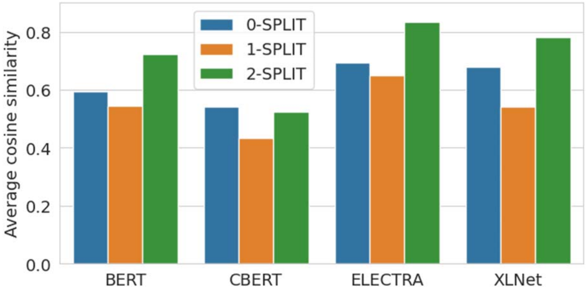

Figure 4: Average similarity values obtained on WiC (BAL) with the AVG strategy.

When it comes to the pooling strategy for representing split-words, AVG is still often the best, but LNG also obtains good results. When comparing instances of the same word, contextual information is more important than word identity, so omitting part of a word does not have such a negative impact as in the inter-word setting.

[15] This is the case in 56% to 60% of split-words in SPLIT-SIM, depending on the subset.

		BERT		CBERT		ELECTRA		XLNet	
		AVG	LM	AVG	LM	AVG	LM	AVG	LM
0-S	SAME	70	**73**	67	**68**	71	71	64	64
	DIFF	**69**	65	**68**	67	**77**	70	60	**62**
1-S	SAME	–	–	–	–	–	–	58	**59**
	DIFF	66	**70**	57	**65**	62	**64**	63	**65**
2-S	SAME	79	**80**	**73**	69	80	80	68	67
	DIFF	**65**	62	58	**60**	**64**	63	73	73

Table 15: Accuracy on WiC pairs with the SAME vs DIFF surface form.

In Table 15, we look at the results of AVG on the original data and when replacing target words with their lemmas (LM) separately on SAME vs DIFF pairs. There is a large gap in accuracy between SAME and DIFF 2-SPLIT pairs, with DIFF pairs obtaining worse results with all models tested[16] except XLNet. 0-SPLIT pairs, on the contrary, are generally less affected by this parameter. While using the lemma is clearly helpful for 1-SPLIT pairs, it does not show a consistent pattern of improvement in the other split-types. We also observe that the average similarities for SAME pairs are higher than for DIFF pairs (e.g., BERT in 0-SPLIT: 0.62 (SAME), 0.54 (DIFF)).

6 Discussion

We have seen that when examined separately, word pairs involving split-words often obtain worse quality similarity estimations than those consisting of full-words; but this depends on the type of word: Split polysemous nouns are better represented than non-split ones. This holds across the models and tokenizers tested, and also when evaluating on words in a narrower frequency range. This shows that word splitting has a negative effect on the representation of many words. We have also seen that in normal conditions, performance on 1-SPLIT is generally the worst one, due mainly to a larger disparity in frequencies of the words in a pair. Our analysis has also confirmed the hypothesis that words that are split in a way that preserves their morphology obtain better quality similarity estimates than words where segmentation splits the word's root(s).

We have noted that similarities for the different split-types are found in different ranges; notably, similarities between two split-words tend to be higher than similarities in 0- and 1-SPLIT pairs. Naturally, this has an effect on the correlation calculated on the full dataset, which is lower than when considering each split-type separately. It would be interesting to develop a similarity measure that allows comparison across split-types, which could rely on information from the rest of the sentence, like BERTScore (Zhang et al., 2020). Another simple way to make similarities comparable would be to bring 2-SPLIT similarities to the 0-SPLIT similarity range by subtracting the average similarity value obtained in 0-SPLIT. The best value to use, however, may vary depending on the application.

One surprising finding relates to the impact of the number of subwords: Similarity estimations are not always more reliable on words involving fewer tokens. This was especially the case for BERT, where we saw that the first token is generally the least useful in building a representation. Given the tendency for the first token to be the longest, this has put the other strategies tested (WAVG and LNG) at a disadvantage.

From our within-word experiments we confirm that word form is reflected in the representations and has a strong impact on similarity, but this does not necessarily mean that comparing words with distinct morphological properties (e.g., singular vs plural) would be detrimental in the inter-word setting. In the within-word setting, SAME pairs compare two equal word forms, whose representation at the initial (static) embedding layer is identical. DIFF pairs, instead, start off with different static embeddings, which results in an overall lower similarity. In SPLIT-SIM, all comparisons are made, by definition, between different words. The fact that two words have different morphological properties may thus have a smaller impact on results.

Most of our findings are consistent between the two kinds of task (inter- and within-word) and across models. One exception is CBERT, which does not assign higher similarities to 2-SPLIT pairs when comparing instances of the same word; and the LNG strategy, which is more useful within-word than inter-word. AVG is, however, the best strategy overall. One direction for future work

[16] A partial explanation is that SAME pairs have a slightly stronger tendency of being T (77% of SAME 2-SPLIT pairs are T, vs 66% of DIFF 2-SPLIT pairs).

would be to find a pooling method that closes the gap in performance between split-types.

Our experiments only involve one language (English), Spearman's correlation, and cosine similarity, although our methodology is not restricted to a single similarity or evaluation metric. Extending this work to more languages is also possible, but less straightforward, due to the need for suitable datasets.

7 Conclusion

We have compared the contextualized representations of words that are segmented into subwords to those of words that have a dedicated embedding in BERT and other models. We have done so through an intrinsic evaluation relying on similarity estimation. Our findings are relevant for any NLP practitioner working with contextualized word representations, and particularly for applications relying on word similarity: (i) Out of the tested strategies for split-word representation, averaging subword embeddings is the best one, with few exceptions; (ii) the quality of split-word representations is often worse than that of full-words, although this depends on the kind of words considered; (iii) similarity values obtained for split-word pairs are generally higher than similarity estimations involving full-words; (iv) the best layers to use differ across split-types; (v) a higher number of tokens does not necessarily, as intuitively thought, decrease representation quality; (vi) in the within-word setting, word form has a negative impact on results when words are split.

Our results also point to specific aspects to which future research and efforts of improvement should be directed. We make our SPLIT-SIM dataset available to facilitate research on split-word representation.

Acknowledgments

We thank the anonymous reviewers and the TACL action editor for their thorough reviews and useful remarks, which helped improve this paper. This research has been supported by the Télécom Paris research chair on Data Science and Artificial Intelligence for Digitalized Industry and Services (DSAIDIS) and by the Agence Nationale de la Recherche, REVITALISE project (ANR-21-CE33-0016).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics. https://doi.org/10.3115/1620754.1620758

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-4010

Carlos S. Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. 2020. CoSimLex: A resource for evaluating graded word similarity in context. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5878–5886, Marseille, France. European Language Resources Association. https://doi.org/10.18653/v1/2020.semeval-1.3

Alexandra Benamar, Cyril Grouin, Meryl Bothua, and Anne Vilnat. 2022. Evaluating tokenizers impact on OOVs representation with transformers models. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4193–4204, Marseille, France. European Language Resources Association.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., Beijing.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146. https://doi.org/10.1162/tacl_a_00051

Rishi Bommasani, Kelly Davis, and Claire Cardie. 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. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.431

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.414

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kenneth Ward Church. 2020. Emerging trends: Subwords, seriously? *Natural Language Engineering*, 26(3):375–382. https://doi.org/10.1017/S1351324920000145

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. 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)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1154

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 *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.609

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2009. Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics. https://doi.org/10.3115/1687878.1687882

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554. https://doi.org/10.1162/COLI_a_00142

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/7287.001.0001

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Christina L. Gagné, Thomas L. Spalding, and Daniel Schmidtke. 2019. LADEC: The large database of English compounds. *Behavior Research Methods*, 51(5):2152–2179. https://doi.org/10.3758/s13428-019-01282-6, PubMed: 31347038

Matthias Gallé. 2019. Investigating the effectiveness of BPE: The power of shorter sequences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1375–1381, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1141

Aina Garí Soler and Marianna Apidianaki. 2021. Let's play mono-poly: BERT can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844. https://doi.org/10.1162/tacl_a_00400

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019. Word usage similarity estimation with sentence representations and automatic substitutes. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 9–21, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-1002

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2022. One word, two sides: Traces of stance in contextualized word representations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3950–3959, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. 2020. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.365

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText corpus.

Aurélie Herbelot and Marco Baroni. 2017. High-risk learning: Acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309, Copenhagen, Denmark. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1030

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695. https://doi.org/10.1162/COLI_a_00237

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In *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)*, pages 3594–3608, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.279

Valentin Hofmann, Hinrich Schütze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.43

Jimin Hong, TaeHee Kim, Hyesu Lim, and Jaegul Choo. 2021. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4692–4700, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.385

Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark. Association for Computational Linguistics. `https://doi.org/10.18653/v1/W17-4706`

Omri Keren, Tal Avinari, Reut Tsarfaty, and Omer Levy. 2022. Breaking character: Are subwords good enough for MRLs after all? *ArXiv*, abs/2204.04748v1.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P18-1007`

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. `https://doi.org/10.18653/v1/D18-2012`

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Explaining and improving BERT performance on lexical semantic change detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2021.eacl-srw.25`

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211. `https://doi.org/10.1037//0033-295X.104.2.211`

Claudia Leacock, Martin Chodorow, and George A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Jiahuan Li, Yutong Shen, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2021a. When is char better than subword: A systematic study of segmentation algorithms for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 543–549, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2021.acl-short.69`

Xiaotao Li, Shujuan You, Yawen Niu, and Wai Chen. 2021b. Learning embeddings for rare words leveraging Internet search engine and spatial location relationships. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 278–287, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2021.starsem-1.26`

Qianchu Liu, Diana McCarthy, and Anna Korhonen. 2020. Towards better context-aware lexical semantics: Adjusting contextualized representations through static anchors. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4066–4075, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.emnlp-main.333`

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language*

Learning, pages 104–113, Sofia, Bulgaria. Association for Computational Linguistics.

Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Thang Vu. 2022. BPE vs. morphological segmentation: A case study on machine translation of four polysynthetic languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 961–971, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.78

Syrielle Montariol and Alexandre Allauzen. 2021. Measure and evaluation of semantic divergence across two languages. In *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)*, pages 1247–1258, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.100

Stephen Mutuvi, Emanuela Boros, Antoine Doucet, Adam Jatowt, Gaël Lejeune, and Moses Odeo. 2022. Fine-tuning de modèles de langues pour la veille épidémiologique multilingue avec peu de ressources (Fine-tuning Language Models for Low-resource Multilingual Epidemic Surveillance). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : Conférence principale*, pages 345–354, Avignon, France. ATALA.

Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.insights-1.1

Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham. 2014. RDRPOSTagger: A ripple down rules-based part-of-speech tagger. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 17–20, Gothenburg, Sweden. Association for Computational Linguistics Linguistics. https://doi.org/10.3115/v1/E14-2005

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. 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)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1128

Mohammad Taher Pilehvar, Dimitri Kartsaklis, Victor Prokhorov, and Nigel Collier. 2018. Card-660: Cambridge rare word dataset - A reliable benchmark for infrequent word representation models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1391–1401, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1169

Tiago Pimentel, Ryan Cotterell, and Brian Roark. 2021. Disambiguatory signals are stronger in word-initial positions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 31–41, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.3

Victor Prokhorov, Mohammad Taher Pilehvar, Dimitri Kartsaklis, Pietro Lio, and Nigel Collier. 2019. Unseen word representation by aligning heterogeneous lexical semantic spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6900–6907. https://doi.org/10.1609/aaai.v33i01.33016900

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*. https://arxiv.org/abs/2003.07082

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? On the monolingual performance of multilingual language models. In *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)*, pages 3118–3135, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.243

Claudia H. Sánchez-Gutiérrez, Hugo Mailhot, S. Hélène Deacon, and Maximiliano A. Wilson. 2018. MorphoLex: A derivational morphological database for 70,000 English words. *Behavior Research Methods*, 50:1568–1580. https://doi.org/10.3758/s13428-017-0981-8 PubMed: 29124719

Timo Schick and Hinrich Schütze. 2020a. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.368

Timo Schick and Hinrich Schütze. 2020b. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8766–8774. https://doi.org/10.1609/aaai.v34i05.6403

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. DWUG: A large resource of diachronic word usage graphs in four languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.567

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE. https://doi.org/10.1109/ICASSP.2012.6289079

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1162

David J. Sheskin. 2003. *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420036268

Robyn Speer. 2022. rspeer/wordfreq: v3.0 (v3.0.2). Zenodo. https://doi.org/10.5281/zenodo.7199437

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.586

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5446

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint:1609.08144*. https://arxiv.org/abs/1609.08144

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics. https://doi.org/10.3115/981732.981751

Chaoqun Yang, Yuanyuan Zhu, Ming Zhong, and Rongrong Li. 2019a. Semantic similarity computation in knowledge graphs: Comparisons and improvements. In *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, pages 249–252. https://doi.org/10.1109/ICDEW.2019.000-5

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019b. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*, pages 19–27, Santiago, Chile. IEEE Computer Society. https://doi.org/10.1109/ICCV.2015.11

A Results with FastText

We choose FastText as a control because of its good results on word similarity, and because it can generate embeddings for all words; 91.8% of all pairs in SPLIT-SIM have both words present in the FastText vocabulary.[17] Table 16 contains the results. The main tendencies observed in Sections 5.1.1 and 5.1.2 are found in these results too: AVG is the best overall strategy and predictions on 1- and 2-SPLIT pairs are almost consistently of lower quality than on 0-SPLIT pairs. We also observe a couple of discrepancies with respect to WUP: Correlations are higher overall, which makes sense as FastText is also a model that learns representations from text and all models (including FastText) have been trained on Wikipedia data. Another important difference is the relative performance of 0-SPLIT and 2-SPLIT in P-N. While with WUP P-N is the only dataset where splitting words is not detrimental to similarity estimation, this is not the case with FastText. However, we note that the difference in performance between 0-SPLIT and 2-SPLIT is much smaller in PN than in the other subsets. This shows that, also in this setting, split polysemous nouns have an advantage with respect to split-words of other types.

[17] The class that is least well represented is 2-SPLIT M-N, but it still has a large majority of in-vocabulary words, with 79% of pairs being completely covered.

		BERT			BERT-FLOTA			CBERT	ELECTRA			XLNet		
		AVG	WAVG	LNG	AVG	WAVG	LNG	–	AVG	WAVG	LNG	AVG	WAVG	LNG
M-N	0-s	65			65			68	68			74		
	1-s	**45***	43*	32*	**39***	37*	31*	55*	**50***	49*	42*	**59***	58*	56*
	2-s	**46***	40*	29*	**36***	**27***	25*	50*	**47***	46*	32*	**50***	49*	43*
M-V	0-s	66			66			65	70			74		
	1-s	**45***	44*	36*	**33***	33*	29*	52*	**51***	50*	43*	**55***	54*	54*
	2-s	**53***	50*	38*	**37***	**34***	26*	55*	**53***	51*	39*	**51***	50*	48*
P-N	0-s	52			52			56	54			60		
	1-s	**40***	39*	30*	**33***	31*	25*	50*	**47***	**48***	41*	**52**	**52**	50
	2-s	**49***	48*	32*	36*	**36***	27*	57	52	**53**	38*	**52**	**52**	46
P-V	0-s	63			63			56	64			66		
	1-s	**46***	45*	37*	**37***	36*	29*	54	**47***	**47***	40*	**55***	**55***	53*
	2-s	**52***	51*	38*	30*	**31***	25*	52*	51*	**52***	38*	50*	**52***	47*

Table 16: Spearman's ρ ($\times$ 100) on SPLIT-SIM (full) using cosine similarities from FastText as a reference.