LEXFIT: Lexical Fine-Tuning of Pretrained Language Models

Ivan Vulić[♠] Edoardo M. Ponti^{♡,♠} Anna Korhonen[♠] Goran Glavaš[◊]

•Language Technology Lab, University of Cambridge, UK

 $^{\heartsuit}$ Mila - Quebec AI Institute and McGill University, Canada

⁽Data and Web Science Group, University of Mannheim, Germany

{iv250,alk23}@cam.ac.uk

edoardo-maria.ponti@mila.quebec

goran@informatik.uni-mannheim.de

Abstract

Transformer-based language models (LMs) pretrained on large text collections implicitly store a wealth of lexical semantic knowledge, but it is non-trivial to extract that knowledge effectively from their parameters. Inspired by prior work on semantic specialization of static word embedding (WE) models, we show that it is possible to expose and enrich lexical knowledge from the LMs, that is, to specialize them to serve as effective and universal "decontextualized" word encoders even when fed input words "in isolation" (i.e., without any context). Their transformation into such word encoders is achieved through a simple and efficient lex*ical fine-tuning procedure* (termed LEXFIT) based on dual-encoder network structures. Further, we show that LEXFIT can yield effective word encoders even with limited lexical supervision and, via cross-lingual transfer, in different languages without any readily available external knowledge. Our evaluation over four established, structurally different lexical-level tasks in 8 languages indicates the superiority of LEXFIT-based WEs over standard static WEs (e.g., fastText) and WEs from vanilla LMs. Other extensive experiments and ablation studies further profile the LEXFIT framework, and indicate best practices and performance variations across LEXFIT variants, languages, and lexical tasks, also directly questioning the usefulness of traditional WE models in the era of large neural models.

1 Introduction

Probing large pretrained encoders like BERT (Devlin et al., 2019) revealed that they contain a wealth of lexical knowledge (Ethayarajh, 2019; Vulić et al., 2020). If type-level word vectors are extracted from BERT with appropriate strategies, they can even outperform traditional word embeddings (WEs) in some lexical tasks (Vulić et al., 2020; Bommasani et al., 2020; Chronis and Erk, 2020). However,

Step 2: Extracting word vectors from (LexFit-ed) BERT



Figure 1: Illustration of the full pipeline for obtaining decontextualized word representations, based on lexically fine-tuning pretrained LMs via dual-encoder networks (Step 1, §2.1), and then extracting the representations from their (fine-tuned) layers (Step 2, §2.2).

both static and contextualized WEs ultimately learn solely from the *distributional* word co-occurrence signal. This source of signal is known to lead to distortions in the induced representations by conflating meaning based on topical relatedness rather than authentic semantic similarity (Hill et al., 2015; Schwartz et al., 2015; Vulić et al., 2017). This also creates a ripple effect on downstream applications, where model performance may suffer (Faruqui, 2016; Mrkšić et al., 2017; Lauscher et al., 2020).

Our work takes inspiration from the methods to correct these distortions and complement the distributional signal with structured information, which were originally devised for static WEs. In particular, the process known as semantic specialization (or retrofitting) injects information about lexical relations from databases like WordNet (Beckwith et al., 1991) or the Paraphrase Database (Ganitkevitch et al., 2013) into WEs. Thus, it accentuates relationships of pure semantic similarity in the refined representations (Faruqui et al., 2015; Mrkšić et al., 2017; Ponti et al., 2019, *inter alia*).

Our goal is to create representations that take advantage of both 1) the expressivity and lexical knowledge already stored in pretrained language models (LMs) and 2) the precision of lexical finetuning. To this effect, we develop LEXFIT, a versatile lexical fine-tuning framework, illustrated in Figure 1, drawing a parallel with universal *sentence* encoders like SentenceBERT (Reimers and Gurevych, 2019).¹ Our working hypothesis, extensively evaluated in this paper, is as follows: pretrained encoders store a wealth of lexical knowledge, but it is not straightforward to extract that knowledge. We can expose this knowledge by *rewiring their parameters* through lexical fine-tuning, and turn the LMs into universal (decontextualized) word encoders.

Compared to prior attempts at injecting lexical knowledge into large LMs (Lauscher et al., 2020), our LEXFIT method is innovative as it is deployed post-hoc on top of already pretrained LMs, rather than requiring joint multi-task training. Moreover, LEXFIT is: 1) more efficient, as it does not incur the overhead of masked language modeling pretraining; and 2) more versatile, as it can be ported to any model independently from its architecture or original training objective. Finally, our results demonstrate the usefulness of LEXFIT: we report large gains over WEs extracted from vanilla LMs and over traditional WE models across 8 languages and 4 lexical tasks, even with very limited and noisy external lexical knowledge, validating the rewiring hypothesis. The code is available at: https://github.com/cambridgeltl/lexfit.

2 From Language Models to (Decontextualized) Word Encoders

The motivation for this work largely stems from the recent work on *probing and analyzing* pretrained language models for various types of knowledge they might implicitly store (e.g., syntax, world knowledge) (Rogers et al., 2020). Here, we focus on their lexical semantic knowledge (Vulić et al., 2020; Liu et al., 2021), with an aim of extracting high-quality static word embeddings from the parameters of the input LMs. In what follows, we describe lexical fine-tuning via dual-encoder networks (§2.1), followed by the WE extraction pro-

cess from the fine-tuned layers of pretrained LMs (§2.2), see Figure 1.

2.1 LEXFIT: Methodology

Our hypothesis is that the pretrained LMs can be turned into effective static decontextualized word encoders via additional inexpensive lexical finetuning (i.e., LEXFIT-ing) on lexical pairs from an external resource. In other words, they can be specialized to encode lexical knowledge useful for downstream tasks, e.g., lexical semantic similarity (Wieting et al., 2015; Mrkšić et al., 2017; Ponti et al., 2018). Let $P = \{(w, v, r)_m\}_{m=1}^M$ refer to the set of M external lexical constraints. Each item $p \in P$ comprises a pair of words w and v, and denotes a semantic relation r that holds between them (e.g., synonymy, antonymy). Further, let P_r denote a subset of P where a particular relation rholds for each item, e.g., P_{syn} is a set of synonymy pairs. Finally, for each *positive* tuple (w, v, r), we can construct 2k negative "no-relation" examples by randomly pairing w with another word $w_{\neg,k'}$, and pairing v with $v_{\neg,k'}$, $k' = 1, \ldots, k$, ensuring that these negative pairs do not occur in P. We refer to the full set of negative pairs as NP. Lexical fine-tuning then leverages P and NP; We propose to tune the underlying LMs (e.g., BERT, mBERT), using external lexical knowledge, via different loss functions, relying on dual-encoder networks with shared LM weights and mean pooling, as illustrated in Figure 1. We now briefly describe several loss functions, evaluated later in §4.

Classification Loss. Similar to prior work on sentence-level text inputs (Reimers and Gurevych, 2019), for each input word pair (w, v) we concatenate their *d*-dimensional encodings w and v (obtained after passing them through BERT and after pooling, see Figure 1) with their element-wise difference |w - v|. The objective is then:

$$\mathcal{L} = \operatorname{softmax} (W(\mathbf{w} \oplus \mathbf{v} \oplus |\mathbf{w} - \mathbf{v}|)).$$
(1)

 \oplus denotes concatenation, and $W \in \mathbb{R}^{3d \times c}$ is a trainable weight matrix of the softmax classifier, where c is the number of classification classes. We experiment with two variants of this objective, termed SOFTMAX henceforth: in the simpler *binary* variant, the goal is to distinguish between positive synonymy pairs (the subset P_{syn}) and the corresponding set of $2k \times |P_{syn}|$ no-relation negative pairs. In the *ternary* variant (c = 3), the classifier must distinguish between synonymy (P_{syn}), and the corresponding set of $2k \times |P_{syn}|$ no-relation negative pairs.

¹These approaches are connected as they are both trained via contrastive learning on dual-encoder architectures, but they provide representations for a different granularity of meaning.

antonyms (P_{ant}) , and no-relation negatives. The classifiers are optimized via standard cross-entropy.

Ranking Loss. The multiple negatives ranking loss (MNEG) is inspired by prior work on learning universal sentence encoders (Cer et al., 2018; Henderson et al., 2019, 2020); the aim of the loss, now adapted to word-level inputs, is to rank true synonymy pairs from P_{syn} over randomly paired words. The similarity between any two words wand v is quantified via the similarity function S operating on their encodings $S(\mathbf{w_i}, \mathbf{w_j})$. In this work we use the scaled cosine similarity following Henderson et al. (2019): $S(\mathbf{w_i}, \mathbf{w_j}) = C \cdot cos(\mathbf{w_1}, \mathbf{w_2})$, where C is the scaling constant. Lexical fine-tuning with MNEG then proceeds in batches of B pairs $(w_i, v_i), \ldots, (w_B, v_B)$ from P_{syn} , with the MNEG loss for a single batch computed as follows:

$$\mathcal{L} = -\sum_{i=1}^{B} S(\mathbf{w}_{i}, \mathbf{v}_{i}) + \sum_{i=1}^{B} \log \sum_{j=1, j \neq i}^{B} e^{S(\mathbf{w}_{i}, \mathbf{v}_{j})}$$
(2)

Effectively, for each batch Eq. (2) maximizes the similarity score of positive pairs (w_i, v_i) , and minimizes the score of B-1 random pairs. For simplicity, as negatives we use all pairings of w_i with v_j -s in the current batch where $(w_i, v_j) \notin P_{syn}$ (Yang et al., 2018; Henderson et al., 2019).

Multi-Similarity Loss. We also experiment with a recently proposed state-of-the-art multi-similarity loss of Wang et al. (2019), labeled MSIM. The aim is again to rank positive examples from P_{syn} above any corresponding no-relation 2k negatives from NP. Again using the scaled cosine similarity scores, the adapted MSIM loss per batch of B positive pairs (w_i , v_i) from P_{syn} is defined as follows:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \left(\log \left(1 + \sum_{k'=1}^{k} e^{C(\cos(\mathbf{w}_{\mathbf{i}}, \mathbf{w}_{\mathbf{i}, \neg, \mathbf{k}'}) - \epsilon)} \right) + \frac{1}{C} \log \left(1 + e^{-C(\cos(\mathbf{w}_{\mathbf{i}}, \mathbf{v}_{\mathbf{i}}) - \epsilon)} \right) \right).$$
(3)

For brevity, in Eq. (3) we only show the formulation with the k negatives associated with $\mathbf{w_i}$, but the reader should be aware that the complete loss function contains another term covering k negatives $v_{i,\neg,k'}$ associated with each $\mathbf{v_i}$. C is again the scaling constant, and ϵ is the offset applied on the similarity matrix.² MSIM can be seen as an extended variant of the MNEG ranking loss. Finally, for any input word w, we extract its word vector via the approach outlined in §2.2; exactly the same approach can be applied to the original LMs (e.g., BERT) or their lexically fine-tuned variants ("LEXFIT-ed" BERT), see Figure 1.

2.2 Extracting Static Word Representations

The extraction of static type-level vectors from any underlying Transformer-based LM, both before and after LEXFIT fine-tuning, is guided by best practices from recent comparative analyses and probing work (Vulić et al., 2020; Bommasani et al., 2020). Starting from an underlying LM with N Transformer layers $\{L_1 \text{ (bottom layer)}, \ldots, L_N \text{ (top)}\}$ and referring to the embedding layer as L_0 , we extract a decontextualized word vector for some input word w, fed into the LM "in isolation" without any surrounding context, following Vulić et al. (2020): 1) w is segmented into 1 or more of its constituent subwords $[sw_i], i \ge 1$, where [] refers to the sequence of i subwords; 2) Special tokens [CLS] and [SEP] are respectively prepended and appended to the subword sequence, and the sequence $[CLS][sw_i][SEP]$ is then passed through the LM; 3) The final representation is constructed as the average over the subword encodings further averaged over $n \leq N$ layers (i.e., all layers up to layer L_n included, denoted as AVG($\leq n$)).³ Further, Vulić et al. (2020) empirically verified that: (a) discarding final encodings of [CLS] and [SEP]produces better type-level vectors - we follow this heuristic in this work; and (b) excluding higher layers from the average may also result in stronger vectors with improved performance in lexical tasks.

This approach operates fully "in isolation" (ISO): we extract vectors of words without any surrounding context. The ISO approach is lightweight: 1) it disposes of any external text corpora; 2) it encodes words efficiently due to the absence of context. Moreover, it allows us to directly study the richness of lexical information stored in the LM's parameters, and to combine it with ISO lexical knowledge from external resources (e.g., WordNet).

3 Experimental Setup

Languages and Language Models. Our language selection for evaluation is guided by the following (partially clashing) constraints (Vulić et al., 2020): a) availability of comparable pretrained monolingual LMs; b) task and evaluation data availabil-

 $^{{}^{2}\}epsilon$ =1; C=20 (also in MNEG). For further technical details we refer the reader to the original paper (Wang et al., 2019).

³Note that this always includes the embedding layer (L_0) .

ity; and c) ensuring some typological diversity of the selection. The final test languages are English (EN), German (DE), Spanish (ES), Finnish (FI), Italian (IT), Polish (PL), Russian (RU), and Turkish (TR). For comparability across languages, we use monolingual uncased BERT Base models for all languages (N = 12 Transformer layers, 12 attention heads, hidden layer dimensionality is 768), available (see the appendix) via the HuggingFace repository (Wolf et al., 2020).

External Lexical Knowledge. We use the standard collection of EN lexical constraints from previous work on (static) word vector specialization (Zhang et al., 2014; Ono et al., 2015; Vulić et al., 2018; Ponti et al., 2018, 2019). It covers the lexical relations from WordNet (Fellbaum, 1998) and Roget's Thesaurus (Kipfer, 2009); it comprises 1,023,082 synonymy (P_{syn}) word pairs and 380,873 antonymy pairs (P_{ant}). For all other languages, we rely on *non-curated noisy* lexical constraints, obtained via an *automatic* word translation method by Ponti et al. (2019); see the original work for the details of the translation procedure.

LEXFIT: Technical Details. The implementation is based on the SBERT framework (Reimers and Gurevych, 2019), using the suggested settings: AdamW (Loshchilov and Hutter, 2018); learning rate of 2e - 5; weight decay rate of 0.01, and we run LEXFIT for 2 epochs. The batch size is 512 with MNEG, and 256 with SOFTMAX and MSIM, where one batch always balances between *B* positive examples and $2k \cdot B$ negatives (see §2.1).

Word Vocabularies and Baselines. We extract decontextualized type-level WEs in each language both from the original BERTs (termed BERT-REG)⁴ and the LEXFIT-ed BERT models for exactly the same vocabulary. Following Vulić et al. (2020), the vocabularies cover the top 100K most frequent words represented in the respective fastText (FT) vectors, trained on lowercased monolingual Wikipedias by Bojanowski et al. (2017).⁵ The equivalent vocabulary coverage allows for a direct comparison of all WEs regardless of the induction/extraction method; this also includes the FT

vectors, used as baseline "traditional" static WEs (termed FASTTEXT.WIKI) in all evaluation tasks.

Evaluation Tasks. We evaluate on the following standard and diverse lexical semantic tasks:

Task 1: Lexical semantic similarity (LSIM) is an established intrinsic task for evaluating static WEs (Hill et al., 2015). We use the recent comprehensive multilingual LSIM benchmark Multi-SimLex (Vulić et al., 2020), which comprises 1,888 pairs in 13 languages, for our EN, ES, FI, PL, and RU LSIM evaluation. We also evaluate on a verbfocused EN LSIM benchmark: SimVerb-3500 (SV) (Gerz et al., 2016), covering 3,500 verb pairs, and SimLex-999 (SL) for DE and IT (999 pairs) (Leviant and Reichart, 2015).⁶

Task 2: Bilingual Lexicon Induction (BLI), a standard task to assess the "semantic quality" of static cross-lingual word embeddings (CLWEs) (Ruder et al., 2019), enables investigations on the alignability of monolingual type-level WEs in different languages before and after the LEXFIT procedure. We learn CLWEs from monolingual WEs obtained with all WE methods using the established and supervision-lenient mapping-based approach (Mikolov et al., 2013a; Smith et al., 2017) with the VECMAP framework (Artetxe et al., 2018). We run main BLI evaluations for 10 language pairs spanning EN, DE, RU, FI, TR.⁷

Task 3: Lexical Relation Prediction (RELP). We assess the usefulness of lexical knowledge in WEs to learn relation classifiers for standard lexical relations (i.e., synonymy, antonymy, hypernymy, meronymy, plus no relation) via a state-ofthe-art neural model for RELP which learns solely based on input type-level WEs (Glavaš and Vulić, 2018). We use the WordNet-based evaluation data of Glavaš and Vulić (2018) for EN, DE, ES; they contain 10K annotated word pairs per language, 8K for training, 2K for test, balanced by class and in the splits. We extract evaluation data for two more languages: FI and IT. We report micro-averaged F_1 scores, averaged across 5 runs for each input WE space; the default RELP model setting is used. In RELP and LSIM, we remove all training and test

⁴For the baseline BERT-REG WEs, we report two variants: (a) *all* performs layerwise averaging over all Transformer layers (i.e., AVG(≤ 12)); (b) *best* reports the peak score when potentially excluding highest layers from the layer averaging (i.e., AVG($\leq n$), $n \leq 12$; see §2.2) (Vulić et al., 2020).

⁵Note that the LEXFIT procedure does not depend on the chosen vocabulary, as it operates only on the lexical items found in the external constraints (i.e., the set P).

⁶The evaluation metric is the Spearman's rank correlation between the average of human LSIM scores for word pairs and the cosine similarity between their respective WEs.

⁷A standard BLI setup and data from Glavaš et al. (2019) is adopted: 5K training word pairs are used to learn the mapping, and another 2K pairs as test data. The evaluation metric is standard Mean Reciprocal Rank (MRR). For EN–ES, we run experiments on MUSE data (Conneau et al., 2018).

RELP/LSIM examples also present in the P_{syn} and P_{ant} sets to avoid any evaluation data leakage.⁸

Task 4: Lexical Simplification (LexSIMP) aims to automatically replace complex words (i.e., specialized terms, less-frequent words) with their simpler in-context synonyms, while retaining grammaticality and conveying the same meaning as the more complex input text (Paetzold and Specia, 2017). Therefore, discerning between semantic similarity (e.g., synonymy injected via LEXFIT) and broader relatedness is critical for LexSIMP (Glavaš and Vulić, 2018). We adopt the standard LexSIMP evaluation protocol used in prior research on static WEs (Ponti et al., 2018, 2019). 1) We use Light-LS (Glavaš and Štajner, 2015), a languageagnostic LexSIMP tool that makes simplifications in an unsupervised way based solely on word similarity in an input (static) WE space; 2) we rely on standard LexSIMP benchmarks, available for EN (Horn et al., 2014), IT (Tonelli et al., 2016), and ES (Saggion, 2017); and 3) we report the standard Accuracy scores (Horn et al., 2014).⁹

Important Disclaimer. We note that the main purpose of the chosen evaluation tasks and experimental protocols is not necessarily achieving state-of-the-art performance, but rather probing the vectors in different lexical tasks requiring different types of lexical knowledge,¹⁰ and offering fair and insightful comparisons between different LEXFIT variants, as well as against standard static WEs (fastText) and non-tuned BERT-based static WEs.

4 Results and Discussion

The main results for all four tasks are summarized in Tables 1-4, and further results and analyses are available in §4.1 (with additional results in the appendix). These results offer multiple axes of comparison, discussed in what follows.

Comparison to Other Static Word Embeddings. The results over all 4 tasks indicate that static WEs from LEXFITed monolingual BERT 1) outperform traditional WE methods such as FT, and 2) offer also large gains over WEs originating from non-LEXFITEd BERTs (Vulić et al., 2020). These results demonstrate that the inexpensive lexical finetuning procedure can indeed turn large pretrained LMs into effective decontextualized word encoders, and this can be achieved for a reasonably wide spectrum of languages for which such pretrained LMs exist. What is more, LEXFIT for all non-EN languages has been run with noisy automatically translated lexical constraints, which holds promise to support even stronger static LEXFITbased WEs with human-curated data in the future, e.g., extracted from multilingual WordNets (Bond and Foster, 2013), PanLex (Kamholz et al., 2014), or BabelNet (Ehrmann et al., 2014).

The results give rise to additional general implications. First, they suggest that the pretrained LMs store even more lexical knowledge than thought previously (Ethayarajh, 2019; Bommasani et al., 2020; Vulić et al., 2020); the role of LEXFIT finetuning is simply to 'rewire' and expose that knowledge from the LM through (limited) lexical-level supervision. To further investigate the 'rewiring' hypothesis, in §4.1, we also run LEXFIT with a drastically reduced amount of external knowledge.

BERT-REG vectors display large gains over FT vectors in tasks such as RELP and LexSIMP, again hinting that plenty of lexical knowledge is stored in the original parameters. However, they still lag FT vectors for some tasks (BLI for all language pairs; LSIM for ES, RU, PL). However, LEXFIT-ed BERT-based WEs offer large gains and outperform FT WEs across the board. Our results indicate that 'classic' WE models such as skip-gram (Mikolov et al., 2013b) and FT are undermined even in their last field of use, lexical tasks.

This comes as a natural finding, given that *word2vec* and FT can in fact be seen as reduced and training-efficient variants of full-fledged language models (Bengio et al., 2003). The modern LMs are pretrained on larger training data with more parameters and with more sophisticated Transformerbased neural architectures. However, it has not been verified before that effective static WEs can be distilled from such LMs. Efficiency differences aside, this begs the following discussion point for future work: with the existence of large pretrained LMs, and effective methods to extract static WEs from them, as proposed in this work, how useful are traditional WE models still in NLP applications?

Lexical Fine-Tuning Objectives. The scores indicate that all LEXFIT variants are effective and can expose the lexical knowledge from the fine-tuned

⁸In BLI and RELP, we do PCA (d = 300) on all input WEs, which slightly improves performance.

⁹For further details regarding the LexSIMP benchmarks and evaluation, we refer the reader to the previous work.

¹⁰RELP and LexSIMP use WEs as *input features* of neural architectures; LSIM and BLI fall under *similarity-based* evaluation tasks (Ruder et al., 2019).

Method	EN	EN: SV	ES	FI	PL	RU	de: SL	it: SL
FASTTEXT.WIKI	44.2	25.8	45.0	58.7	36.7	35.8	41.3	30.5
BERT-REG (all)	46.7	23.9	42.4	55.3	32.0	30.6	31.3	28.8
BERT-REG (best)	51.8	28.9	44.2	61.5	32.4	30.7	34.6	31.1
MNEG [113 min]	73.6	<u>68.3</u>	62.3	72.0	52.4	50.4	49.7	58.7
MSIM [174 min]	74.3	69.6	<u>61.8</u>	<u>71.1</u>	<u>51.8</u>	<u>49.9</u>	49.7	58.9
SOFTMAX (binary) [177 min]	64.3	58.8	58.9	<u>62.4</u>	44.7	44.6	43.7	49.4
SOFTMAX (ternary) [212 min]	67.8	61.7	59.4	<u>66.2</u>	46.3	38.8	45.3	52.4

Table 1: Results in the LSIM task; Spearman's ρ correlation scores (× 100). k = 1 for the MSIM and SOFTMAX lexical fine-tuning variants (see §3). SV = SimVerb-3500; SL = SimLex-999. The best score in each column is in bold; the second best is underlined. Additional LSIM results are available in the appendix. The numbers in [] denote the average fine-tuning time with each LEXFIT objective per 1 epoch in English (1 GTX TITAN X GPU).

Method	EN-DE	EN-TR	EN-FI	EN-RU	DE-TR	DE-FI	DE-RU	TR–FI	TR-RU	FI–RU	avg
FASTTEXT.WIKI	61.0	43.3	48.8	52.2	35.8	43.5	46.9	35.8	36.4	<u>43.9</u>	44.8
BERT-REG (all)	44.6	37.9	47.1	47.3	32.3	39.5	41.2	35.2	31.9	38.7	39.6
BERT-REG (best)	47.2	39.0	48.6	48.8	32.3	39.5	41.2	35.2	31.9	39.2	40.3
MNEG	58.1	46.2	57.7	54.0	<u>36.2</u>	<u>46.1</u>	$ \frac{46.7}{46.7} \\ 43.5 \\ 44.6 $	39.6	<u>36.7</u>	42.4	<u>46.4</u>
MSIM	<u>58.9</u>	<u>45.9</u>	57.7	<u>53.7</u>	37.1	46.4		<u>39.4</u>	37.4	44.2	46.7
SOFTMAX (binary)	57.9	<u>45.3</u>	53.8	53.6	35.9	44.3		38.4	36.0	42.8	45.2
SOFTMAX (ternary)	57.1	44.9	54.8	52.7	35.2	44.0		38.4	34.9	41.1	44.8

Table 2: Results in the BLI task (MMR \times 100). k = 1. Additional BLI results are available in the appendix.

Method	EN	DE	ES	FI	IT
FASTTEXT.WIKI	66.0	60.1	62.2	68.2	64.8
BERT-REG (all)	71.4	67.3	65.1	69.6	66.8
BERT-REG (best)	71.8	67.9	65.5	69.9	67.2
MNEG	74.1	<u>69.7</u>	67.8	71.3	71.1
MSIM	74.3	69.0	<u>68.6</u>	72.2	<u>71.4</u>
SOFTMAX (binary)	74.0	68.4	67.4	71.5	70.1
SOFTMAX (ternary)	75.5	70.3	70.3	73.2	71.3

Table 3: Results in the RELP task (Micro- $F_1 \times 100$, averaged over 5 runs). More results in the appendix.

Method	EN	ES	IT
FASTTEXT.WIKI	11.4	16.3	14.2
BERT-REG (all)	71.6	38.3	32.7
MNEG MSIM SOFTMAX (binary) SOFTMAX (ternary)	83.8 <u>84.4</u> 84.8 84.0	55.3 56.7 56.7 53.9	45.0 45.4 45.8 44.2

Table 4: LexSIMP results (Accuracy $\times 100$).

BERTs. However, there are differences across their task performance: the ranking-based MNEG and MSIM variants display stronger performance on similarity-based ranking lexical tasks such as LSIM and BLI. The classification-based SOFTMAX objective is, as expected, better aligned with the RELP task, and we note slight gains with its ternary variant which leverages extra antonymy knowledge. This finding is well aligned with the recent findings demonstrating that task-specific pretraining results in stronger (sentence-level) task performance (Glass et al., 2020; Henderson et al., 2020; Lewis et al., 2020). In our case, we show that task-specific lexical fine-tuning can reshape the underlying LM's parameters to not only act as a *universal word encoder*, but also towards a *particular lexical task*.

The per-epoch time measurements from Table 1 validate the efficiency of LEXFIT as a post-training fine-tuning procedure. Previous approaches that attempted to inject lexical information (i.e., word senses and relations) into large LMs (Lauscher et al., 2020; Levine et al., 2020) relied on *joint* LM (re)training *from scratch*: it is effectively costlier than training the original BERT models.

Performance across Languages and Tasks. As expected, the scores in absolute terms are highest for EN: this is attributed to (a) larger pretraining LM data as well as (b) to clean external lexical knowledge. However, we note encouragingly large gains in target languages even with noisy translated lexical constraints. LEXFIT variants show similar relative patterns across different languages and tasks. We note that, while BERT-REG vectors are unable to match FT performance in the BLI task, our LEXFIT methods (e.g., see MNEG and MSIM BLI scores) outperform FT WEs in this task



Figure 2: Varying the amount of external lexical knowledge for LEXFIT (MSIM, k = 1).



Figure 3: Impact of the number of of negative examples k on lexical task performance. In the legends, A = MSIM; B = SOFTMAX (the binary variant plotted for RELP and BLI, ternary for RELP). The numbers in the parentheses denote performance of FT vectors. The full results with more languages and LEXFIT variants are in the appendix.

as well, offering improved alignability (Søgaard et al., 2018) between monolingual WEs. The large gains of BERT-REG over FT in RELP and LexSIMP across all evaluation languages already suggest that plenty of lexical knowledge is stored in the pretrained BERTs' parameters; however, LEXFIT-ing the models offers further gains in LexSIMP and RELP across the board, even with limited external supervision (see also Figure 2c).

High scores with FI in LSIM and BLI are aligned with prior work (Virtanen et al., 2019; Rust et al., 2021) that showcased strong monolingual performance of FI BERT in sentence-level tasks. Along this line, we note that the final quality of LEXFITbased WEs in each language depends on several factors: 1) pretraining data; 2) the underlying LM; 3) the quality and amount of external knowledge.

4.1 Further Discussion

The multi-component LEXFIT framework allows for a plethora of additional analyses, varying components such as the underlying LM, properties of the LEXFIT variants (e.g., negative examples, finetuning duration, the amount of lexical constraints). We now analyze the impact of these components on the "lexical quality" of the LEXFIT-tuned static WEs. Unless noted otherwise, for computational feasibility and to avoid clutter, we focus 1) on a subset of target languages: EN, ES, FI, IT, 2) on the MSIM variant (k = 1), which showed robust perfor-



Figure 4: Performance comparison between languagespecific monolingual BERT models (MONOBERT) and mBERT serving as the underlying LM. MSIM (k = 1).

mance in the main experiments before, and **3**) on LSIM, BLI, and RELP as the main tasks in these analyses, as they offer a higher language coverage.

Varying the Amount of Lexical Constraints. We also probe what amount of lexical knowledge is required to turn BERTs into effective decontextualized word encoders by running tests with reduced lexical sets P sampled from the full set. The scores over different P sizes, averaged over 5 samples per each size, are provided in Figure 2, and we note that they extend to other evaluation languages and LEXFIT objectives. As expected, we do observe performance drops with fewer external data. However, the decrease is modest even when relying on

	n =	2	4	6	8	10	12
	EN:REG		51.8				
LSIM	EN:MSIM FI:REG FI:MSIM	57.3	61.5 59.8 64.1	61.5	61.1	59.3	55.3
DLI	EN-FI:REG		43.8				
BLI	EN-FI:MSIM	40.2	45.6	50.7	54.3	56.1	57.7

Table 5: Task performance of WEs extracted via layerwise averaging over different Transformer layers (AVG($\leq n$) extraction variants; §2.2) for a selection of tasks and languages. LEXFIT variant: MSIM (k = 1). REG = BERT-REG. Highest scores per row are in bold.

only 5k external constraints (e.g., see the scores in BLI and RELP for all languages; EN Multi-SimLex score is 69.4 with 50k constraints, 65.0 with 5k), or even non-existent (RELP in FI).

Remarkably, the LEXFIT performance with only 10k or 5k fine-tuning pairs¹¹ remains substantially higher than with FT or BERT-REG WEs in all tasks. This empirically validates LEXFIT's sample efficiency and further empirically corroborates our *knowledge rewiring* hypothesis: the original LMs already contain plenty of useful lexical knowledge implicitly, and even a small amount of external supervision can expose that knowledge.

Copying or Rewiring Knowledge? Large gains over BERT-REG even with mere 5k pairs (LEXFITing takes only a few minutes), where the large portion of the 100K word vocabulary is not covered in the external input, further reveal that LEXFIT does not only copy the knowledge of seen words and relations into the LM: it leverages the (small) external set to generalize to uncovered words.

We confirm this hypothesis with another experiment where our input LM is the same BERT Base architecture parameters with the same subword vocabulary as English BERT, but with its parameters now *randomly* initialized using the Xavier initialization (Glorot and Bengio, 2010). Running LEX-FIT on this model for 10 epochs with the full set of lexical constraints (see §3) yields the following LSIM scores: 23.1 (Multi-SimLex) and 14.6 (SimVerb), and the English RELP accuracy score of 61.8%. The scores are substantially higher than those of fully random static WEs (see also the appendix), which indicates that the LEXFIT procedure does enable storing some lexical knowledge into the model parameters. However, at the same

¹¹When sampling all reduced sets, we again deliberately excluded all words occurring in our LSIM benchmarks.

time, these scores are substantially lower than the ones achieved when starting from LM-pretrained models, even when LEXFIT is run with mere 5k fine-tuning lexical pairs.¹² This again strongly suggests that LEXFIT 'unlocks' already available lexical knowledge stored in the pretrained LM, yielding benefits beyond the knowledge available in the external data. Another line of recent work (Liu et al., 2021) further corroborates our findings.

Multilingual LMs. Prior work indicated that massively multilingual LMs such as multilingual BERT (mBERT) (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) cannot match the performance of their language-specific counterparts in both lexical (Vulić et al., 2020) and sentence-level tasks (Rust et al., 2021). We also analyze this conjecture by LEXFIT-ing mBERT instead of monolingual BERTs in different languages. The results with MSIM (k = 1) are provided in Figure 4; we observe similar comparison trends with other languages and LEXFIT variants, not shown due to space constraints. While LEXFIT-ing mBERT offers huge gains over the original mBERT model, sometimes even larger in relative terms than with monolingual BERTs (e.g., LSIM scores for EN increase from 0.21 to 0.69, and from 0.24 to 0.60 for FI; BLI scores for EN-FI rise from 0.21 to 0.37), it cannot match the absolute performance peaks of LEXFIT-ed monolingual BERTs.

Storing the knowledge of 100+ languages in its limited parameter budget, mBERT still cannot capture monolingual knowledge as accurately as language-specific BERTs (Conneau et al., 2020). However, we believe that its performance with LEXFIT may be further improved by leveraging recently proposed multilingual LM adaptation strategies that mitigate a mismatch between shared multilingual and language-specific vocabularies (Artetxe et al., 2020; Chung et al., 2020; Pfeiffer et al., 2020); we leave this for future work.

Layerwise Averaging. A consensus in prior work (Tenney et al., 2019; Ethayarajh, 2019; Vulić et al., 2020) points that out-of-context lexical knowledge in pretrained LMs is typically stored in bottom Transformer layers (see Table 5). However, Table 5 also reveals that this does not hold after LEXFIT-ing: the tuned model requires knowledge from all layers to extract effective decontextualized WEs and reach peak task scores. Effectively, this means

¹²The same findings hold for other tasks and languages.

that, through lexical fine-tuning, model "reformats" all its parameter budget towards storing useful lexical knowledge, that is, it specializes as (decontextualized) word encoder.

Varying the Number of Negative Examples and their impact on task performance is recapped in Figure 3b. Overall, increasing k does not benefit (and sometimes even hurts) performance – the exceptions are EN LSIM; and the RELP task with the SOFTMAX variant for some languages. We largely attribute this to the noise in the target-language lexical pairs: with larger k values, it becomes increasingly difficult for the model to discern between noisy positive examples and random negatives.

Longer Fine-Tuning. Instead of the standard setup with 2 epochs (see §3), we run LEXFIT for 10 epochs. The per-epoch snapshots of scores are summarized in the appendix. The scores again validate that LEXFIT is sample-efficient: longer fine-tuning yields negligible to zero improvements in EN LSIM and RELP after the first few epochs, with very high scores achieved after epoch 1 already. It even yields small drops for other languages in LSIM and BLI: we again attribute this to slight overfitting to noisy target-language lexical knowledge.

5 Conclusion and Future Work

We proposed LEXFIT, a lexical fine-tuning procedure which transforms pretrained LMs such as BERT into effective decontextualized word encoders through dual-encoder architectures. Our experiments demonstrated that the lexical knowledge already stored in pretrained LMs can be further exposed via additional inexpensive LEXFITing with (even limited amounts of) external lexical knowledge. We successfully applied LEXFIT even to languages without any external human-curated lexical knowledge. Our LEXFIT word embeddings (WEs) outperform "traditional" static WEs (e.g., fastText) across a spectrum of lexical tasks across diverse languages in controlled evaluations, thus directly questioning the practical usefulness of the traditional WE models in modern NLP.

Besides inducing better static WEs for lexical tasks, following the line of lexical probing work (Ethayarajh, 2019; Vulić et al., 2020), our goal in this work was to understand how (and how much) lexical semantic knowledge is coded in pretrained LMs, and how to 'unlock' the knowledge from the LMs. We hope that our work will be beneficial for all lexical tasks where static WEs from traditional

WE models are still largely used (Schlechtweg et al., 2020; Kaiser et al., 2021).

Despite the extensive experiments, we only scratched the surface, and can indicate a spectrum of future enhancements to the proof-of-concept LEXFIT framework beyond the scope of this work. We will test other dual-encoder loss functions, including finer-grained relation classification tasks (e.g., in the SOFTMAX variant), and hard (instead of random) negative examples (Wieting et al., 2015; Mrkšić et al., 2017; Lauscher et al., 2020; Kalantidis et al., 2020). While in this work, for simplicity and efficiency, we focused on fully decontextualized ISO setup (see §2.2), we will also probe alternative ways to extract static WEs from pretrained LMs, e.g., averages-over-context (Liu et al., 2019; Bommasani et al., 2020; Vulić et al., 2020). We will also investigate other approaches to procuring more accurate external knowledge for LEXFIT in target languages, and extend the framework to more languages, lexical tasks, and specialized domains. We will also focus on reducing the gap between pretrained monolingual and multilingual LMs.

Acknowledgments

We thank the three anonymous reviewers, Nils Reimers, and Jonas Pfeiffer for their helpful comments and suggestions. Ivan Vulić and Anna Korhonen are supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no. 648909) awarded to Korhonen, and the ERC PoC Grant MultiConvAI: Enabling Multilingual Conversational AI (no. 957356). Goran Glavaš is supported by the Baden Württemberg Stiftung (Eliteprogramm, AGREE grant).

References

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of ACL 2018*, pages 789–798.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of ACL* 2020, pages 4623–4637.
- Richard Beckwith, Christiane Fellbaum, Derek Gross, and George A. Miller. 1991. WordNet: A lexical database organized on psycholinguistic principles. *Lexical acquisition: Exploiting on-line resources to build a lexicon*, pages 211–231.

- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Re*search, 3:1137–1155.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.
- Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of ACL 2020*, pages 4758–4781.
- Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of ACL 2013*, pages 1352–1362.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In Proceedings of EMNLP 2018, pages 169–174.
- Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? When it's like a rabbi! multi-prototype bert embeddings for estimating semantic relationships. In *Proceedings of CoNLL 2020*, page 227–244.
- Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of EMNLP 2020*, pages 4536–4546.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL 2020*, pages 8440–8451.
- Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of ICLR 2018*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.
- Maud Ehrmann, Francesco Cecconi, Daniele Vannella, John Philip McCrae, Philipp Cimiano, and Roberto Navigli. 2014. Representing multilingual data as linked data: The case of BabelNet 2.0. In *Proceedings of LREC 2014*, pages 401–408.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of EMNLP-IJCNLP 2019*, pages 55– 65.

- Manaal Faruqui. 2016. *Diverse Context for Learning Word Representations*. Ph.D. thesis, Carnegie Mellon University.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL-HLT 2015*, pages 1606–1615.
- Christiane Fellbaum. 1998. WordNet. MIT Press.
- Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT 2013*, pages 758–764.
- Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A largescale evaluation set of verb similarity. In *Proceedings of EMNLP 2016*, pages 2173–2182.
- Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Shrivatsa Bhargav, Dinesh Garg, and Avirup Sil. 2020. Span selection pretraining for question answering. In *Proceedings of* ACL 2020, pages 2773–2782.
- Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of ACL-IJCNLP 2015*, pages 63–68.
- Goran Glavaš and Ivan Vulić. 2018. Discriminating between lexico-semantic relations with the specialization tensor model. In *Proceedings of NAACL-HLT* 2018, pages 181–187.
- Goran Glavaš and Ivan Vulić. 2018. Explicit retrofitting of distributional word vectors. In *Proceedings of ACL 2018*, pages 34–45.
- Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate crosslingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of ACL 2019*, pages 710–721.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS 2010*, pages 249–256.
- Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of EMNLP 2020*, pages 2161–2174.
- Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *Proceedings of ACL 2019*, pages 5392–5404.

- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using Wikipedia. In *Proceedings of ACL 2014*, pages 458–463.
- Jens Kaiser, Sinan Kurtyigit, Serge Kotchourko, and Dominik Schlechtweg. 2021. Effects of pre- and post-processing on type-based embeddings in lexical semantic change detection. In *Proceedings of EACL 2021*, pages 125–137.
- Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. In Proceedings of NeurIPS 2020.
- David Kamholz, Jonathan Pool, and Susan M. Colowick. 2014. PanLex: Building a resource for panlingual lexical translation. In *Proceedings of LREC* 2014, pages 3145–3150.
- Barbara Ann Kipfer. 2009. Roget's 21st Century Thesaurus (3rd Edition). Philip Lief Group.
- Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2020. Specializing unsupervised pretraining models for word-level semantic similarity. In *Proceedings of COLING 2020*, pages 1371–1383.
- Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *CoRR*, abs/1508.00106.
- Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of ACL 2020*, pages 4656–4667.
- Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. *CoRR*, abs/2006.15020.
- Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective and self-supervised: Transforming masked language models into universal lexical and sentence encoders. *CoRR*, abs/2104.08027.
- Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of CoNLL* 2019, pages 33–43.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proceedings of ICLR 2018*.

- Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint, CoRR*, abs/1309.4168.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS 2013*, pages 3111–3119.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of ACL 2017*, pages 1777–1788.
- Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the ACL*, 5:309–324.
- Masataka Ono, Makoto Miwa, and Yutaka Sasaki. 2015. Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of NAACL-HLT 2015*, pages 984–989.
- Gustavo Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. UNKs everywhere: Adapting multilingual language models to new scripts. *CoRR*, abs/2012.15562.
- Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Adversarial propagation and zero-shot cross-lingual transfer of word vector specialization. In *Proceedings of EMNLP 2018*, pages 282–293.
- Edoardo Maria Ponti, Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Cross-lingual semantic specialization via lexical relation induction. In *Proceedings of EMNLP 2019*, pages 2206–2217.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERTnetworks. In *Proceedings of EMNLP 2019*, pages 3982–3992.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: what we know about how BERT works. *Transactions of the ACL*, 8:842–866.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.
- 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 ACL* 2021.

- Horacio Saggion. 2017. Automatic text simplification. Synthesis Lectures on Human Language Technologies, 10(1):1–137.
- Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. Semeval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of SemEval 2020*, pages 1–23.
- Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings* of *CoNLL* 2015, pages 258–267.
- Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR 2017*.
- Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of ACL 2018*, pages 778–788.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of ACL 2019*, pages 4593–4601.
- Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. 2016. SIMPITIKI: A simplification corpus for Italian. In *Proceedings of CLiC-IT 2016*.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *CoRR*, abs/1912.07076.
- Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart, and Anna Korhonen. 2020. Multi-Simlex: A largescale evaluation of multilingual and cross-lingual lexical semantic similarity. *Computational Linguistics*.
- Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen. 2018. Post-specialisation: Retrofitting vectors of words unseen in lexical resources. In *Proceedings of NAACL-HLT 2018*, pages 516–527.
- Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of EMNLP 2020*, pages 7222–7240.
- Ivan Vulić, Roy Schwartz, Ari Rappoport, Roi Reichart, and Anna Korhonen. 2017. Automatic selection of context configurations for improved class-specific word representations. In *Proceedings of CoNLL* 2017, pages 112–122.

- Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of CVPR 2019*, pages 5022–5030.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the ACL*, 3:345–358.
- 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 EMNLP 2020: System Demonstrations*, pages 38–45.
- Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-Yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In Proceedings of The 3rd Workshop on Representation Learning for NLP, pages 164–174.
- Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. 2014. Word semantic representations using bayesian probabilistic tensor factorization. In *Proceedings of EMNLP 2014*, pages 1522–1531.

Language	URL
EN	https://huggingface.co/bert-base-uncased
DE	https://huggingface.co/bert-base-german-dbmdz-uncased
ES	https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased
FI	https://huggingface.co/TurkuNLP/bert-base-finnish-uncased-v1
IT	https://huggingface.co/dbmdz/bert-base-italian-xxl-uncased
PL	https://huggingface.co/dkleczek/bert-base-polish-uncased-v1
RU	https://huggingface.co/DeepPavlov/rubert-base-cased
TR	https://huggingface.co/dbmdz/bert-base-turkish-uncased
Multilingual	https://huggingface.co/bert-base-multilingual-uncased

Table 6: URLs of the pretrained LMs used in our study, obtained via the HuggingFace repo (Wolf et al., 2020).

Method	EN	EN: SV	ES	FI	RU	de: SL
FASTTEXT.WIKI	44.2	25.8	45.0	58.7	35.8	41.3
BERT-REG (all)	46.7	23.9	42.4	55.3	30.6	31.3
BERT-REG (best)	51.8	28.9	44.2	61.5		34.6
MNEG		(a. a	<i>(</i>))			40 -
-	73.6	68.3	62.3	72.0	50.4	49.7
MSIM	74.2	(0.((1.0	71.1	40.0	40.7
$\begin{array}{l} k = 1 \\ k = 2 \end{array}$	74.3 74.3	69.6 69.6	61.8 61.8	$\frac{71.1}{71.1}$	$\frac{49.9}{49.9}$	49.7 49.6
k = 2 k = 4			61.9	$\frac{71.1}{68.4}$	$\frac{49.9}{48.6}$	47.9
k = 4 k = 8	<u>75.7</u> 75.9	$\frac{71.7}{72.3}$	<u>62.0</u>	66.4	49.9	46.5
SOFTMAX (binary)			<u></u>			
k = 1	64.3	58.8	58.8	62.4	44.6	43.7
k = 2	67.9	61.4	60.1	67.6	46.6	45.9
k = 4	70.2	64.9	60.6	69.6	46.7	47.0
k = 8	71.3	67.2	61.4	70.2	46.7	47.6
SOFTMAX (ternary)						
k = 1	67.8	61.7	59.4	66.2	38.8	45.3
k = 2	68.8	62.6	60.1	66.7	42.4	46.6
k = 4	70.6	65.8	59.7	67.8	45.3	47.6
k = 8	71.6	67.8	60.9	68.5	45.0	47.0

Table 7: A summary of results in the lexical semantic similarity (LSIM) task (Spearman's ρ correlation scores), also showing the dependence on the number of negative examples per positive example: *k*. The scores for EN, ES, FI, and RU are reported on the Multi-SimLex lexical similarity benchmark (Vulić et al., 2020) (1,888 word pairs). The scores for DE, not represented in Multi-SimLex, are calculated on a smaller benchmark: German SimLex-999 (Hill et al., 2015; Leviant and Reichart, 2015) (SL; 999 word pairs) For EN, we also report the scores on the verb similarity dataset SimVerb-3500 (Gerz et al., 2016) (SV). All LEXFIT-based WEs have been induced from "lexically fine-tuned" LMs, relying on the standard setup described in §3, and relying on lexical constraints also summarized in §3. All results with LEXFIT variants are obtained relying on the best-performing configuration for extracting word representations from the comparative study of Vulić et al. (2020). BERT-REG denotes the extraction of word representations (again with the best strategy from prior work) from the regular underlying BERT models, which were not further "LEXFIT-ed": (all) layerwise averaging over all Transformer layers; (best) the highest results reported by Vulić et al. (2020), often achieved by excluding several highest layers from the layerwise averaging. The highest scores per column are in **bold**; the second best result per column is <u>underlined</u>.



Figure 5: Impact of LEXFIT fine-tuning duration (i.e., the number of fine-tuning epochs) in three lexical tasks (LSIM, BLI, RELP). We report a subset of results with a selection of languages and language pairs, relying on the MSIM (k = 1) LEXFIT fine-tuning variant. Similar trends are observed with other LEXFIT variants.

		(a) Tr	aining di	ctionary:	5,000 wor	d transla	tion pairs				
Method	EN-DE	EN-TR	EN-FI	EN-RU	DE-TR	DE-FI	DE-RU	TR-FI	TR-RU	FI–RU	avg
FASTTEXT.WIKI	61.0	43.3	48.8	52.2	35.8	43.5	46.9	35.8	36.4	<u>43.9</u>	44.8
BERT-REG (all)	44.6	37.9	47.1	47.3	32.3	39.5	41.2	35.2	31.9	38.7	39.6
MNEG											
-	58.1	46.2	57.7	54.0	36.2	46.1	<u>46.7</u>	39.6	36.7	42.4	46.4
MSIM											
k = 1	<u>58.9</u>	<u>45.9</u>	<u>57.6</u>	<u>53.7</u>	37.1	46.4	<u>46.7</u>	<u>39.4</u>	37.4	44.2	46.7
k = 2	57.2	44.4	56.7	52.8	35.7	44.7	46.1	39.3	37.4	42.2	45.7
k = 4	57.0	43.6	55.2	51.5	35.5	43.6	44.8	38.0	35.1	39.3	44.4
k = 8	55.4	44.0	53.0	49.1	34.0	41.8	42.2	36.5	32.0	37.5	42.6
SOFTMAX (binary)											
k = 1	57.9	45.3	53.8	53.6	35.9	44.3	43.5	38.4	36.0	42.8	45.2
k = 2	55.8	44.6	55.4	51.9	34.7	43.8	41.9	39.1	34.6	40.0	44.2
k = 4	55.8	43.8	54.9	51.4	34.6	42.8	39.9	37.9	33.3	39.0	43.3
k = 8	54.2	43.1	54.4	50.2	33.3	42.0	39.7	36.8	32.9	38.7	42.5
SOFTMAX (ternary)											
k = 1	57.1	44.9	54.8	52.7	35.2	44.0	44.6	38.4	34.9	41.1	44.8
$\tilde{k} = 2$	55.7	45.2	54.4	53.2	34.1	43.6	42.6	38.4	34.5	40.7	44.2
$\tilde{k} = 4$	55.5	44.7	55.1	52.6	34.0	42.8	40.2	38.6	33.4	40.7	43.8
$\tilde{k} = 8$	54.9	44.2	53.3	51.5	33.3	41.3	38.7	37.2	32.9	37.8	42.5
		(b) Tr	aining di	ctionary:	1,000 wor	d transla	tion pairs				
Method											
	EN-DE	EN-TR	EN-FI	EN-RU	DE-TR	DE-FI	DE-RU	TR-FI	TR-RU	FI–RU	avg
FASTTEXT.WIKI	EN-DE 53.9	EN-TR 31.7	EN-FI 35.4	EN-RU 39.0	DE-TR 23.0	DE-FI 31.5	DE–RU 37.8	TR-FI 21.4	TR-RU 22.2	FI-RU 29.6	avg 32.6
FASTTEXT.WIKI	53.9 26.4	31.7 20.6	35.4 25.8	39.0 25.4	23.0 17.4	31.5 24.6	37.8 23.4	21.4 20.4	22.2 15.6	29.6 21.4	32.6 22.1
FASTTEXT.WIKI BERT-REG (all)	53.9	31.7	35.4	39.0	23.0	31.5	37.8	21.4	22.2	29.6	32.6
FASTTEXT.WIKI BERT-REG (all) MNEG	53.9 26.4 55.2	31.7 20.6 34.1	35.4 25.8 44.8	39.0 25.4 40.3	23.0 17.4 <u>25.9</u>	31.5 24.6 33.9	37.8 23.4 <u>31.7</u>	21.4 20.4 29.2	22.2 15.6 22.3	29.6 21.4 <u>30.1</u>	32.6 22.1 34.8
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1	53.9 26.4 55.2 54.3	31.7 20.6 34.1 33.2	35.4 25.8 <u>44.8</u> 45.1	39.0 25.4 40.3 39.3	23.0 17.4 <u>25.9</u> 26.0	31.5 24.6 33.9 33.9	37.8 23.4 <u>31.7</u> 31.4	21.4 20.4 29.2 29.1	22.2 15.6 22.3 23.8	29.6 21.4 <u>30.1</u> 30.8	32.6 22.1 34.8 34.7
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM	53.9 26.4 55.2 54.3 54.3	31.7 20.6 34.1 <u>33.2</u> <u>32.0</u>	35.4 25.8 44.8	39.0 25.4 40.3 39.3 38.8	23.0 17.4 <u>25.9</u> 26.0 24.6	31.5 24.6 33.9 32.7	37.8 23.4 <u>31.7</u> 31.4 30.0	21.4 20.4 29.2 <u>29.1</u> 28.4	22.2 15.6 22.3 23.8 22.1	29.6 21.4 <u>30.1</u>	32.6 22.1 34.8 <u>34.7</u> 33.3
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1	53.9 26.4 55.2 54.3 54.3 53.0	31.7 20.6 34.1 <u>33.2</u> <u>32.0</u> 31.8	35.4 25.8 44.8 45.1 43.3 41.6	39.0 25.4 40.3 39.3 38.8 38.1	23.0 17.4 25.9 26.0 24.6 24.3	31.5 24.6 33.9 32.7 30.9	37.8 23.4 <u>31.7</u> 31.4 30.0 27.4	21.4 20.4 29.2 29.1 28.4 25.2	22.2 15.6 22.3 23.8 22.1 20.1	29.6 21.4 30.1 30.8 27.1 26.1	32.6 22.1 34.8 <u>34.7</u> 33.3 31.9
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2	53.9 26.4 55.2 54.3 54.3	31.7 20.6 34.1 <u>33.2</u> <u>32.0</u>	35.4 25.8 44.8 45.1 43.3	39.0 25.4 40.3 39.3 38.8	23.0 17.4 <u>25.9</u> 26.0 24.6	31.5 24.6 33.9 32.7	37.8 23.4 <u>31.7</u> 31.4 30.0	21.4 20.4 29.2 <u>29.1</u> 28.4	22.2 15.6 22.3 23.8 22.1	29.6 21.4 <u>30.1</u> 30.8 27.1	32.6 22.1 34.8 <u>34.7</u> 33.3
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4	53.9 26.4 55.2 54.3 53.0 51.4	31.7 20.6 34.1 33.2 32.0 31.8 30.6	35.4 25.8 44.8 45.1 43.3 41.6 40.1	39.0 25.4 40.3 39.3 38.8 38.1 36.4	23.0 17.4 25.9 26.0 24.6 24.3 22.7	31.5 24.6 33.9 32.7 30.9 28.7	37.8 23.4 <u>31.7</u> 31.4 30.0 27.4 24.9	21.4 20.4 29.2 29.1 28.4 25.2 24.2	22.2 15.6 22.3 23.8 22.1 20.1 17.8	29.6 21.4 30.1 30.8 27.1 26.1 23.7	32.6 22.1 34.8 <u>34.7</u> 33.3 31.9 30.1
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1	31.7 20.6 34.1 <u>33.2</u> 32.0 31.8 30.6 32.0	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4	39.0 25.4 40.3 39.3 38.8 38.1 36.4 39.7	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6	31.5 24.6 33.9 32.7 30.9 28.7 32.1	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4	32.6 22.1 34.8 <u>34.7</u> 33.3 31.9 30.1 33.5
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary)	53.9 26.4 55.2 54.3 53.0 51.4	31.7 20.6 34.1 33.2 32.0 31.8 30.6	35.4 25.8 44.8 45.1 43.3 41.6 40.1	39.0 25.4 40.3 39.3 38.8 38.1 36.4	23.0 17.4 25.9 26.0 24.6 24.3 22.7	31.5 24.6 33.9 32.7 30.9 28.7	37.8 23.4 <u>31.7</u> 31.4 30.0 27.4 24.9	21.4 20.4 29.2 29.1 28.4 25.2 24.2	22.2 15.6 22.3 23.8 22.1 20.1 17.8	29.6 21.4 30.1 30.8 27.1 26.1 23.7	32.6 22.1 34.8 <u>34.7</u> 33.3 31.9 30.1
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3 52.7	31.7 20.6 34.1 <u>33.2</u> 32.0 31.8 30.6 32.0	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4	39.0 25.4 40.3 39.3 38.8 38.1 36.4 39.7	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5 25.5	31.5 24.6 33.9 32.7 30.9 28.7 32.1	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8 32.7
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1 k = 2	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3	31.7 20.6 34.1 3 <u>3.2</u> 32.0 31.8 30.6 32.0 32.3	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4 43.7	39.0 25.4 40.3 39.3 38.8 38.1 36.4 <u>39.7</u> <u>39.6</u>	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5	31.5 24.6 33.9 32.7 30.9 28.7 32.1 <u>33.4</u>	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6 31.3 31.3	21.4 20.4 29.2 29.1 28.4 25.2 24.2 24.2 27.2 28.9	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1 22.8	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4 27.8	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1 k = 2 k = 4	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3 52.7	31.7 20.6 34.1 3 <u>3.2</u> 32.0 31.8 30.6 32.0 32.3 31.9	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4 43.7 42.4	39.0 25.4 40.3 39.3 38.8 38.1 36.4 <u>39.7</u> <u>39.6</u> 37.4	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5 25.5	31.5 24.6 33.9 32.7 30.9 28.7 32.1 33.4 32.2	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6 31.3 29.6	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2 28.9 27.5	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1 22.8 21.0	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4 27.8 26.5	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8 32.7
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1 k = 2 k = 4 k = 2 k = 4 k = 8	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3 52.7	31.7 20.6 34.1 3 <u>3.2</u> 32.0 31.8 30.6 32.0 32.3 31.9	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4 43.7 42.4	39.0 25.4 40.3 39.3 38.8 38.1 36.4 <u>39.7</u> <u>39.6</u> 37.4	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5 25.5	31.5 24.6 33.9 32.7 30.9 28.7 32.1 33.4 32.2	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6 31.3 29.6	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2 28.9 27.5	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1 22.8 21.0	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4 27.8 26.5	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8 32.7
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1 k = 2 k = 4 k = 8 SOFTMAX (ternary)	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3 52.7 52.2	31.7 20.6 34.1 <u>33.2</u> 32.0 31.8 30.6 32.0 32.3 31.9 31.1 32.0	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4 43.7 42.1	39.0 25.4 40.3 39.3 38.8 38.1 36.4 <u>39.7</u> 39.6 37.4 38.0 38.7	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5 25.5 23.5 24.2	31.5 24.6 33.9 32.7 30.9 28.7 32.1 33.4 32.2 30.0	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6 31.3 29.6 28.4 28.4	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2 28.9 27.5 25.9	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1 22.8 21.0 20.8 22.0	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4 27.8 26.5 25.8	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8 32.7 31.8 33.2
FASTTEXT.WIKI BERT-REG (all) MNEG - MSIM k = 1 k = 2 k = 4 k = 8 SOFTMAX (binary) k = 1 k = 2 k = 4 k = 8 SOFTMAX (ternary) k = 1	53.9 26.4 55.2 54.3 54.3 53.0 51.4 54.1 52.3 52.7 52.2 53.5	31.7 20.6 34.1 33.2 32.0 31.8 30.6 32.0 32.3 31.9 31.1	35.4 25.8 44.8 45.1 43.3 41.6 40.1 40.4 43.7 42.4 42.1 42.8	39.0 25.4 40.3 39.3 38.8 38.1 36.4 39.7 39.6 37.4 38.0	23.0 17.4 25.9 26.0 24.6 24.3 22.7 25.6 25.5 25.5 23.5	31.5 24.6 33.9 32.7 30.9 28.7 32.1 <u>33.4</u> 32.2 30.0 32.0	37.8 23.4 31.7 31.4 30.0 27.4 24.9 31.6 31.3 29.6 28.4 31.0	21.4 20.4 29.2 29.1 28.4 25.2 24.2 27.2 28.9 27.5 25.9 27.6	22.2 15.6 22.3 23.8 22.1 20.1 17.8 23.1 22.8 21.0 20.8	29.6 21.4 30.1 30.8 27.1 26.1 23.7 29.4 27.8 26.5 25.8 28.6	32.6 22.1 34.8 34.7 33.3 31.9 30.1 33.5 33.8 32.7 31.8

Table 8: Results in the BLI task across different language pairs and dual-encoder lexical fine-tuning (LEXFIT) objectives (MNEG, MSIM, SOFTMAX). The size of the training dictionary is (a) 5,000 or (b) 1,000 word translation pairs. MRR scores reported; **avg** refers to the average score across all 10 language pairs. All results with LEXFIT variants are obtained relying on the best-performing configuration for extracting word representations from the comparative study of Vulić et al. (2020). BERT-REG denotes the extraction of word representations (again with the best strategy from prior work) from the regular underlying BERT models, which were not further "LEXFIT-ed": (all) layerwise averaging over all Transformer layers. The highest scores per column for each training dictionary size are in **bold**; the second best result is <u>underlined</u>.

Method	EN	DE	ES	FI
RANDOM.XAVIER	$47.3_{\pm 0.3}$	$51.2_{\pm 0.8}$	$49.7_{\pm 0.9}$	$51.8_{\pm 0.5}$
FASTTEXT.WIKI	$66.0_{\pm 0.8}$	$60.1_{\pm 0.7}$	$62.2_{\pm 1.6}$	$68.2_{\pm 0.3}$
BERT-REG (all)	$71.4_{\pm 1.2}$	$67.3_{\pm 0.3}$	$65.1_{\pm 1.1}$	$69.6_{\pm 0.6}$
BERT-REG (best)	$71.8_{\pm 0.2}$	$67.9_{\pm 0.8}$	$65.5_{\pm 1.2}^{-}$	$69.9_{\pm 0.5}$
MNEG				
-	$74.1_{\pm 1.1}$	$69.7_{\pm 1.0}$	$67.8_{\pm 0.3}$	$71.3_{\pm 1.5}$
MSIM				
k = 1	$74.3_{\pm 1.3}$	$69.0_{\pm 0.6}$	$68.6_{\pm 0.7}$	$72.2_{\pm 0.4}$
k = 2	$73.8_{\pm 0.8}$	$68.6_{\pm 1.2}$	$68.4_{\pm 0.5}$	$72.3_{\pm 0.3}$
k = 4	$73.5_{\pm 1.0}$	$68.8_{\pm 1.2}$	$67.1_{\pm 1.1}$	$72.0_{\pm 0.9}$
k = 8	$72.1_{\pm 0.9}$	$68.9_{\pm 0.6}$	$67.6_{\pm 1.3}$	$71.2_{\pm 1.3}$
SOFTMAX (binary)				
k = 1	$74.0_{\pm 1.6}$	$68.4_{\pm 0.7}$	$67.4_{\pm 0.3}$	$71.5_{\pm 0.6}$
k = 2	$73.8_{\pm 1.0}$	$69.4_{\pm 0.5}$	$67.4_{\pm 0.8}$	$71.2_{\pm 0.9}$
k = 4	$73.9_{\pm 1.0}$	$69.4_{\pm 0.9}$	$67.2_{\pm 1.1}$	$72.7_{\pm 0.7}$
k = 8	$73.2_{\pm 1.0}$	$68.2_{\pm 1.1}$	$67.8_{\pm 1.1}$	$71.4_{\pm 1.1}$
SOFTMAX (ternary)				
k = 1	$75.5_{\pm 0.5}$	$70.3_{\pm 0.7}$	$70.3_{\pm 0.6}$	$73.2_{\pm 1.2}$
k = 2	$\overline{75.7_{\pm 0.8}}$	$68.8_{\pm 0.8}$	$69.8_{\pm 0.8}$	$73.2_{\pm 0.5}$
k = 4	$74.4_{\pm 0.8}$	$69.9_{\pm 0.5}$	$69.9_{\pm 0.5}$	$72.2_{\pm 0.6}$
k = 8	$74.0_{\pm 0.2}$	$\overline{68.1_{\pm 0.9}}$	$\overline{68.1_{\pm 0.9}}$	$72.3_{\pm 0.4}$

Table 9: A summary of results in the relation prediction (RELP) task, also showing the dependence on the number of negative examples per positive example: k. Micro-averaged F_1 scores, obtained as averages over 5 experimental runs for each input word vector space; standard deviation is also reported in the subscript. RANDOM.XAVIER are 768-dimensional vectors for the same vocabularies, randomly initialized via Xavier initialization (Glorot and Bengio, 2010). The highest scores per column are in **bold**, the second best is <u>underlined</u>.