XL-LEXEME: WiC Pretrained Model for Cross-Lingual LEXical sEMantic changE

Pierluigi Cassotti, Lucia Siciliani, Marco de Gemmis, Giovanni Semeraro
and **Pierpaolo Basile**
University of Bari Aldo Moro
{firstname.lastname}@uniba.it

Abstract

The recent introduction of large-scale datasets for the WiC (Word in Context) task enables the creation of more reliable and meaningful contextualized word embeddings. However, most of the approaches to the WiC task use cross-encoders, which prevent the possibility of deriving comparable word embeddings. In this work, we introduce XL-LEXEME, a Lexical Semantic Change Detection model. XL-LEXEME extends SBERT, highlighting the target word in the sentence. We evaluate XL-LEXEME on the multilingual benchmarks for SemEval-2020 Task 1 - Lexical Semantic Change (LSC) Detection and the RuShiftEval shared task involving five languages: English, German, Swedish, Latin, and Russian. XL-LEXEME outperforms the state-of-the-art in English, German and Swedish with statistically significant differences from the baseline results and obtains state-of-the-art performance in the RuShiftEval shared task.

1 Introduction and Motivation

Lexical Semantic Change (LSC) Detection is the task of automatically identifying words that change their meaning over time. The LSC Detection task implicitly aims to disambiguate synchronic word sense occurrences and then find differences in the word sense frequencies in different periods. Word Sense Disambiguation (WSD) is a long-studied task in Natural Language Processing (Navigli, 2009), which consists of associating the correct sense to a word occurring in a specific context. WSD involves some crucial issues, such as relying on a fixed sense inventory. Fixed sense inventories ignore the diachronic aspect of language because they can miss older unused senses or be outdated and missing new senses.

The Word in Context task (WiC) (Pilehvar and Camacho-Collados, 2019) aims to overcome these issues. In this work, we train a model on the WiC task and then use it to perform LSC Detection. In the WiC task, given the word w and two different contexts $C1$, $C2$, the systems have to determine whether the meaning of w is the same in the two contexts or not. Our approach is grounded on the assumption that models trained on the WiC tasks are robust enough to transfer the knowledge learned in a synchronic setting to a diachronic one. We summarise the main contribution of this work as follows: (i) We propose a pre-trained bi-encoder model, called XL-LEXEME, on a large-scale dataset for the WiC task, which allows us to obtain comparable lexical-based representations; (ii) We assert the effectiveness of XL-LEXEME despite the computational limitation compared to the cross-encoder architecture for the LSC Detection task; (iii) Experiments on the LSC Detection task show that XL-LEXEME outperforms state-of-the-art LSC Detection models for English, German, Swedish, and Russian.

2 Related Work

LSC Detection systems can be categorized based on the distributional embeddings used to tackle the LSC Detection task. One category is represented by those approaches that adopt type-base (i.e., static) embeddings. UWB (Prazák et al., 2020; Prazák et al., 2021) represents an example of this category of systems. First, it employs word2vec Skip-gram with Negative Sampling (Mikolov et al., 2013) to compute a semantic space for each corpus. It uses techniques like the Canonical Correlation Analysis (Hardoon et al., 2004) and the Orthogonal Transformation (Hamilton et al., 2016) to align the abovementioned spaces. Therefore, the cosine similarity between the vectors representing the word in two different spaces is used to detect the semantic shift.

With the increasing use of contextualized word embeddings, numerous approaches employing BERT-base models have been developed for LSC Detection (Montanelli and Periti, 2023; Laicher

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1577–1585

et al., 2021). In TempoBERT (Rosin et al., 2022), the authors exploit the concept of Masked Language Modeling (MLM), where the goal is to train a language model to predict a masked portion of text given the remaining part. In particular, they employ this technique to encode the concept of time into a BERT model. This is done by concatenating a specific token representing time to the text sequence. At inference time, TempoBERT can be used to predict the year of a sentence, masking the time reference, or to predict a masked token of the sentence conditioned by the time reference. In the same line of research, in Temporal Attention (Rosin and Radinsky, 2022), the authors investigate the effect of modifying the model instead of the input sentence like in TempoBERT. This is done by extending the model's attention mechanism to consider the time when computing the weight of each word. The time dimension is encoded using a different query embedding matrix for each timestamp.

Another kind of approach exploits the information coming from other tasks to perform LSC Detection. GlossReader represents an example (Rachinskiy and Arefyev, 2021), where a model based on XML-R (Conneau et al., 2020b) is first trained on English SemCor (Miller et al., 1994) with glosses from WordNet 3.0 (Miller, 1992) to perform WSD. Exploiting the zero-shot cross-lingual characteristics of XML-R, the authors used the same model to perform LSC Detection in the Russian language. With DeepMistake (Arefyev et al., 2021), the authors take advantage of the WiC task instead of WSD. They train a cross-encoder with XML-R as an underlying Language Model on the MCL-WiC training and development set and fine-tune on the RuSemShift dataset (Rodina and Kutuzov, 2020). DeepMistake, differently from XL-LEXEME, relies on the cross-encoder architecture and exploits only the MCL-WiC training dataset.

3 XL-LEXEME

Generally, for pairwise sentence similarity tasks, BERT models use a cross-encoder, in which the pairwise sequences are jointly encoded, and the overall vectors are used for the classification. However, in several tasks, the cross-encoder is not suitable since it cannot provide a distinct meaningful representation for each sentence. An approach to overcome this issue involves pooling the BERT output encoded vectors, which often results in worse performance. Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) overcomes the limitation of cross-encoders using a Siamese Network, i.e., the weights of the underlying networks are shared. SBERT encodes the two sequences separately in the BERT model exploiting the Siamese architecture. The sequence-level representation is obtained by averaging the output encoded vectors, which are directly compared using similarity measures such as cosine similarity.

Meanwhile, cross-encoders perform better since they are trained to profit from the attention over the whole input. In this work, we introduce XL-LEXEME[1] which mirrors models for pairwise sequence similarity tasks and adapts them to the WiC task, giving prominence to the target word, i.e. the word for which we want to detect the LSC. The model takes as input two sequences s_1 and s_2. The sequences are tokenized using subwords tokenizer, such as Sentence Piece (Kudo and Richardson, 2018), and the special tokens <t> and </t> are used as target word delimiters (Xie et al., 2021):

$$\begin{aligned} s_1 &= w_1, ..., \texttt{<t>}, w_i^t, ..., w_{i+k}^t, \texttt{</t>}, ..., w_N \\ s_2 &= w_1, ..., \texttt{<t>}, w_j^t, ..., w_{j+p}^t, \texttt{</t>}, ..., w_M \end{aligned} \quad (1)$$

where N and M represent the number of subwords of the sequence s_1 and s_2 respectively, while $w_i^t, ..., w_{i+k}^t$ and $w_j^t, ..., w_{j+p}^t$ are the subwords of the target words. In the following, we describe the baseline cross-encoder and XL-LEXEME based on a bi-encoder. For the cross-encoder, the two input sequences are concatenated by the special token $[SEP]$ in an overall sequence $s = [CLS]\ s_1\ [SEP]\ s_2\ [SEP]$. If the length of s, i.e. $N + M + 3$, is greater than the maximum sequence length λ, then the sequence s is cut such that the length of s_1 and s_2 is less than $\lambda^* = \frac{\lambda-3}{2}$. To comply with the maximum length, the left and right contexts of the sequence are truncated. For instance, s_1 is truncated as follows:

$$s_1 = w_{n_0}, ..., \texttt{<t>}, w_i^t, ..., w_{i+k}^t, \texttt{</t>}, ..., w_{n_1} \quad (2)$$

where $n_0 = \max(0, i - 1 - \frac{\lambda^*-k-2}{2})$ and $n_1 = \min(N, i + k + 1 + \frac{\lambda^*-k-2}{2})$. The truncated sequence has a length $\gamma < \lambda$. The encoded representations of each subword $(v_1, v_2, ..., v_\gamma)$ are

[1]The XL-LEXEME code is available on GitHub https://github.com/pierluigic/xl-lexeme. The XL-LEXEME model is available in the Hugging Face Model Hub https://huggingface.co/pierluigic/xl-lexeme.

summed to get the encoded representation of the overall sequence, i.e. $s^{enc} = \sum_i^{\gamma} v_i$. Finally, the vector s^{enc} is used to compute the logits:

$$logit = \log \sigma(W s^{enc}) \quad (3)$$

where $W \in \mathbb{R}^{1 \times d}$. The model is trained to minimize the Binary Cross-entropy loss function.

XL-LEXEME is a bi-encoder that encodes the input sequences using a Siamese Network into two different vector representations. Each sequence is tokenized and truncated according to the maximum length λ^*, using Equation (2). We thus obtain the new lengths γ_1, γ_2. The vector representation is computed as the sum of the encoded subwords $(v_1, v_2, ..., v_\gamma)$, i.e. $s_1^{enc} = \sum_i^{\gamma_1} v_i$ and $s_2^{enc} = \sum_j^{\gamma_2} v_j$.

XL-LEXEME is trained to minimize the Contrastive loss (Hadsell et al., 2006):

$$\ell = \frac{1}{2}\left[y \cdot \delta^2 + (1-y) \cdot \max(0, m-\delta)^2\right] \quad (4)$$

where we adopt a margin $m = 0.5$. We use as default distance δ the cosine distance between the encoded representations of s_1 and s_2, i.e. $\delta = \cos(s_1^{enc}, s_2^{enc})$. The main advantage of XL-LEXEME concerning models based on the cross-encoder architecture is efficiency. The time cost can be directly derived from the different architectures that exploit XL-LEXEME and the cross-encoder baseline. The self-attention time complexity $O(N^2 * d)$ depends on the vector dimension d and the sequence length, which is N for the cross-encoder and $\frac{N}{2}$ for XL-LEXEME. For XL-LEXEME, the time complexity is reduced to $O((\frac{N}{2})^2 * 2d)$.

4 Experimental setting

4.1 Lexical Semantic Change Detection

SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection (Schlechtweg et al., 2020) is the first task on Unsupervised Lexical Semantic Change Detection in English, German, Swedish, and Latin languages. For each language, two corpora represent two different periods (T0, T1). Moreover, a set of target words, annotated using the DUREL framework (Schlechtweg et al., 2018), are provided. SemEval-2020 Task 1 involves two subtasks. The binary classification task requires assigning a label (changed/stable) to each target word. The ranking task sorts the target words according to their degree of semantic change. In this work, we focus on Subtask 2, and for the sake of simplicity, we refer to SemEval-2020 Task 1 Subtask 2 as SemEval-2020 Task 1.

RuShiftEval, different from SemEval-2020 Task 1, involves three sub-corpora extracted from the Russian National Corpus spanning three periods. Models are evaluated on the resulting three test sets, namely RuShiftEval1 (pre-Soviet and Soviet), RuShiftEval2 (Soviet and post-Soviet), and RuShiftEval3 (pre-Soviet and post-Soviet). RuShiftEval provides participants with development data that can be used for tuning models. RuShiftEval aims to corroborate if training data can improve LSC Detection models. The development data rely on the RuSemShift dataset (Rodina and Kutuzov, 2020), which includes two sets of 70 target words for the pre-Soviet to Soviet period and Soviet to post-Soviet period, respectively. The dataset also includes annotated pairwise sentences, which can be used for training the models.

4.2 Training details

XL-LEXEME and the cross-encoder are trained using XLM-RoBERTa (XLM-R) (Conneau et al., 2020a) large as the underlying Language Model[2] and using an NVIDIA GeForce RTX 3090. As for training data, the model uses the training data of MCL-WiC (Martelli et al., 2021), AM^{2}ICO (Liu et al., 2021), and XL-WiC datasets (Raganato et al., 2020) merged with the randomly sampled 75% of the respective development data of each dataset. The remaining 25% of the development data is used to fine-tune hyper-parameters. Moreover, we augment training data for the cross-encoder by swapping the order of sentences in the training set (Martelli et al., 2021).

We use AdamW optimizer and linear learning warm-up over the 10% of training data. We perform a grid search for the hyper-parameters optimization, tuning the learning rate in {1e-6, 2e-6, 5e-6, 1e-5, 2e-5} and the weight decay {0.0, 0.01}. Table 3 (Appendix A) shows the selected hyper-parameters. We sample 200 sentences containing the target word for each language and each period. The sampling is repeated ten times, and the results are averaged over the ten iterations. We use the same methodology of Rachinskiy and Arefyev (2021) for sampling sentences from the RuShiftEval corpora. We sample sentences in which we find the exact match with the target words with no pre-

[2]The XLM-R model is fine-tuned during the training.

processing of the SemEval dataset. The LSC score is computed as the average distance between the vectors over the two different periods:

$$\text{LSC}(s^{t_0}, s^{t_1}) = \frac{1}{N \cdot M} \sum_{i=0}^{N} \sum_{j=0}^{M} \delta(s_i^{t_0}, s_j^{t_1}) \quad (5)$$

where δ is the distance measure, i.e. $\delta = 1 - \log \sigma(W s^{enc})$ for the cross-encoder baseline and $\delta = \cos(s_1^{enc}, s_2^{enc})$ for XL-LEXEME.

5 Results

Table 1 and Table 2 report the results on the SemEval-2020 Task 1 Subtask 2 and the results on the RuShiftEval test set. The results of the best systems are in bold. XL-LEXEME achieve the best score for English, German, Swedish, RuShiftEval1, RuShiftEval2, and RuShiftEval3. XL-LEXEME achieves a strong Spearman correlation for English and Swedish languages and a solid correlation on the German dataset, obtaining a significative correlation ($p < 0.001$). XL-LEXEME obtains no significant results in the Latin language since the predicted scores for the target words are not correlated with the test set. Latin is underrepresented in the training data of XLM-R, and there are no similar languages in the WiC dataset that we use for training XL-LEXEME. Moreover, the Latin dataset is more challenging as it involves the first corpus written in ancient Latin, which differs in many aspects from modern Latin. For this reason, XL-LEXEME could be ineffective in ancient languages and, in general, in languages that are not widely covered by the WiC dataset.

We report the statistical significance of the difference between the performance of XL-LEXEME concerning the other models. The statistical significance of the difference is computed using Fisher's z-transformation (Press, 2002). XL-LEXEME obtains stronger correlations than the cross-encoder, but the differences are not significant. The correlations obtained on the English and the German datasets are significantly different ($p < 0.05$) for all the systems that participated in the SemEval-2020 Task 1 but not for TempoBERT and Temporal Attention. On the other side, TempoBERT and Temporal Attention obtain a Spearman correlation on English and German that is not statistically different from the systems on the SemEval-2020 Task 1 leaderboard. In the Swedish language, XL-LEXEME is the only one obtaining a significantly different correlation from the Count baseline results. XL-LEXEME showed its effectiveness also in Swedish, although the WiC dataset does not cover this language. Presumably, Swedish benefits from the presence of other languages descending from the Old Norse language, namely Danish and Norwegian.

XL-LEXEME obtains competitive results for the Russian language in the RuShiftEval leaderboard. Contrary to XL-LEXEME, Deep Mistake and Gloss Reader are fine-tuned on the RuSemShift dataset. The differences between XL-LEXEME and the best two systems in the leaderboard are not statically significant. Moreover, in Table 2, the results of XL-LEXEME fine-tuned on the RuSemShift are shown. Although the fine-tuned model achieves the best correlation scores in the three datasets, the difference between DeepMistake and GlossReader is not significant.

6 Conclusion

In this work, we introduced XL-LEXEME, a model for LSC Detection. XL-LEXEME is pre-trained on a large WiC dataset to mirror sentence-level encoders focusing on specific words in contexts. We evaluated our model on two Lexical Semantic Change Detection datasets: SemEval-2020 Task 1 and RuShiftEval. XL-LEXEME outperforms state-of-the-art models for LSC Detection in English, German, Swedish, and Russian datasets, with significant differences from the baselines. The XL-LEXEME effectiveness and efficiency make it reliable for LSC Detection on large diachronic corpora.

7 Limitations

While the vector representations obtained using XL-LEXEME for different languages are potentially comparable, lying on the same geometric space, the evaluation of cross-lingual semantic changes cannot be performed for lacking cross-lingual LSC Detection resources. SemEval 2020 Task 1 datasets consist of small sets of target words, i.e., the number of target words for English, German, Latin, and Swedish is 37, 48, 40, and 31, respectively. The example of the Latin language highlights that XL-LEXEME can perform poorly on languages that are underrepresented in the training set of XLM-R and not covered by the WiC dataset. Generally, at the moment is not possible to state precisely how and how much XL-LEXEME

	SemEval-2020 Task 1 Subtask 2 Leaderboard						Temporal BERT		cross-encoder	XL-LEXEME
Lang.	UG_Student _Intern	Jiaxin & Jinan	cs2020	UWB	Count baseline	Freq. baseline	TempoBERT	Temporal Attention		
EN	0.422	0.325	0.375	0.367	0.022	-0.217	0.467	†0.520	†0.752	**0.757**
DE	0.725	0.717	0.702	0.697	0.216	0.014	-	†0.763	†0.837	**0.877**
SV	†0.547	†0.588	†0.536	†0.604	-0.022	-0.150	-	-	†0.680	**0.754**
LA	0.412	**0.440**	0.399	0.254	0.359	†0.020	0.512	0.565	†0.016	-0.056
Avg.	0.527	0.518	0.503	0.481	0.144	-0.083	-	-	0.571	**0.583**

Table 1: Results (Spearman correlation) on the SemEval-2020 Task 1 Subtask 2 test set. The symbol † indicates there is no statistical difference with the correlation obtained by XL-LEXEME.

	RuShiftEval Leaderboard				cross-encoder	XL-LEXEME	XL-LEXEME (Fine-tuned)
Dataset	GlossReader	DeepMistake	UWB	Baseline			
RuShiftEval1	†0.781	†0.798	0.362	0.314	†0.727	0.775	**0.799**
RuShiftEval2	†0.803	†0.773	0.354	0.302	†0.753	0.822	**0.833**
RuShiftEval3	†0.822	†0.803	0.533	0.381	†0.748	0.809	**0.842**
Avg.	0.802	0.791	0.417	0.332	0.743	0.802	**0.825**

Table 2: Results (Spearman correlation) on the RuShiftEval test set. The symbol † indicates there is no statistical difference with the correlation obtained by XL-LEXEME.

performance is affected by the language distribution in the XLM-R training set and the WiC dataset.

Acknowledgements

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

This work has in part been funded by the research program Change is Key! supported by Riksbankens Jubileumsfond (under reference number M21-0021).

References

Nikolay Arefyev, Daniil Homskiy, Maksim Fedoseev, Adis Davletov, Vitaly Protasov, and Alexander Panchenko. 2021. DeepMistake: Which Senses are Hard to Distinguish for a WordinContext Model. In *Computational Linguistics and Intellectual Technologies - Papers from the Annual International Conference "Dialogue" 2021*, volume 2021-June. Section: 20.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12):2639–2664.

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.

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.

Qianchu Liu, Edoardo Maria Ponti, Diana McCarthy, Ivan Vulic, and Anna Korhonen. 2021. AM2iCo: Evaluating Word Meaning in Context across Low-Resource Languages with Adversarial Examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7151–7162. Association for Computational Linguistics.

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 2021. SemEval-2021 Task 2: Multilingual and Cross-lingual Word-in-Context Disambiguation (MCL-WiC). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36, Online. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*.

George A. Miller. 1992. WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a Semantic Concordance for Sense Identification. In *Human Language Technology, Proceedings of a Workshop held at Plainsboro, New Jerey, USA, March 8-11, 1994*. Morgan Kaufmann.

Stefano Montanelli and Francesco Periti. 2023. A survey on contextualised semantic shift detection. *arXiv preprint arXiv:2304.01666*.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Comput. Surv.*, 41(2).

Mohammad Taher Pilehvar and José 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1267–1273. Association for Computational Linguistics.

Ondrej Prazák, Pavel Pribán, and Stephen Taylor. 2021. UWB@ RuShiftEval Measuring Semantic Difference as per-word Variation in Aligned Semantic Spaces. In *Computational Linguistics and Intellectual Technologies - Papers from the Annual International Conference "Dialogue" 2021*, volume 2021-June. Section: 20.

Ondrej Prazák, Pavel Pribán, Stephen Taylor, and Jakub Sido. 2020. UWB at SemEval-2020 Task 1: Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING2020*, pages 246–254. International Committee for Computational Linguistics.

William H. Press. 2002. *Numerical recipes in C++: the art of scientific computing, 2nd Edition (C++ ed., print. is corrected to software version 2.10)*. Cambridge University Press.

Maxim Rachinskiy and Nikolay Arefyev. 2021. Zeroshot Crosslingual Transfer of a Gloss Language Model for Semantic Change Detection. In *Computational Linguistics and Intellectual Technologies - Papers from the Annual International Conference "Dialogue" 2021*, volume 2021-June. Section: 20.

Alessandro Raganato, Tommaso Pasini, José Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A Multilingual Benchmark for Evaluating Semantic Contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7193–7206. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1037–1047, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2022. Time Masking for Temporal Language Models. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 833–841. ACM.

Guy D. Rosin and Kira Radinsky. 2022. Temporal Attention for Language Models. *CoRR*, abs/2202.02093.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, SemEval@COLING2020*, pages 1–23. International Committee for Computational Linguistics.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A Framework for the Annotation

of Lexical Semantic Change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.

Shuyi Xie, Jian Ma, Haiqin Yang, Lianxin Jiang, Yang Mo, and Jianping Shen. 2021. PALI at SemEval-2021 Task 2: Fine-Tune XLM-RoBERTa for Word in Context Disambiguation. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 713–718, Online. Association for Computational Linguistics.

A Hyper-parameters

Hyper-parameter	**Value**
hidden act	gelu
hidden dropout prob	0.1
hidden size	1024
initializer range	0.02
intermediate size	4096
layer norm eps	1e-05
max position embeddings	514
num attention heads	16
num hidden layers	24
position embedding type	absolute
vocab size	250004
learning rate	
cross-encoder	1e-05
XL-LEXEME	1e-05
weight decay	
cross-encoder	0.01
XL-LEXEME	0.00
max sequence length	
cross-encoder	$\lambda = 256$
XL-LEXEME	$\lambda* = 128$

Table 3: XL-LEXEME and cross-encoder hyper-parameters.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3 and Section 4

☑ B1. Did you cite the creators of artifacts you used?
References

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4 and References

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3 and Section 4

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3 and Section 4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C ☑ Did you run computational experiments?

Section 3 and Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 and Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.