The NTNU System in MultiGED-2023: Contextual Flair Embeddings for Multilingual Grammatical Error Detection

Lars Bungum and **Björn Gambäck**
Department of Computer Science
NTNU, Trondheim, Norway
{lars.bungum,gamback}@ntnu.no

Arild Brandrud Næss
NTNU Business School
NTNU, Trondheim, Norway
arild.naess@ntnu.no

Abstract

The paper presents a monolithic approach to grammatical error detection, which uses one model for all languages, in contrast to the individual approach, which creates separate models for each language. For both approaches, pre-trained embeddings are the only external knowledge sources. Two sets of embeddings (Flair and BERT) are compared as well as two approaches to the problem of multilingual rammar detection, building individual and monolithic systems for multilingual grammar error detection. The system submitted to the test phase of the MultiGED-2023 shared task ranked 5th of 6 systems. In the subsequent open phase, more experiments were conducted, improving results. These results show the individual models to perform better than the monolithic ones and BERT embeddings working better than Flair embeddings for the individual models, while the picture is more mixed for the monolithic models.

1 Introduction

The MultiGED-2023 shared task on Multilingual Grammatical Error Detection (MGED; Volodina et al., 2023) presents six datasets, in the languages Czech, German, Italian, and Swedish as well as two in English; all well-resourced languages with more than 10 million speakers. Although not strictly required, the task did encourage the submission of multilingual systems. This work compares both approaches, multilingual and individual models for each language.

The NTNU system aimed to answer two research questions with its submission:

(i) the feasibility of using Flair embeddings (Akbik et al., 2018) provided by the FlairNLP framework (Akbik et al., 2019a) vs. the more traditional BERT embeddings, and

(ii) the impact of training RNNs using language-specific and multilingual embeddings, respectively, to address the problem.

Consequently, no other external resources — or synthetic data — were used. The submission to the test phase of the shared task was a multilingual system, which ranked 5th of 6 systems.

The rest of the paper is structured as follows: first, Section 2 discusses relevant background, and Section 3 briefly describes the dataset. Section 4 outlines the proposed method and Section 5 presents the results, while Section 6 provides a discussion. Finally, Section 7 concludes and outlines ideas for future work.

2 Background

Grammatical error detection (GED) has received increased attention in the research community. Figure 1 shows the number of publications about GED registered in the Web of Science[1] over the last 31 years, most of which are categorized as computer science disciplines. The results were obtained by searching for the query "Grammatical Error Detection" and asking for a citation report, from which the chart was downloaded at the time of submission.

Bryant et al. (2023) summarized the state-of-the-art of the closely related field of grammati-

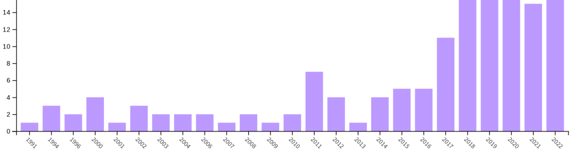


Figure 1: Number of GED publications registered in the Web of Science per year from 1991 (1) to 2022 (27).

[1] http://www.webofscience.com

cal error correction (GEC) as of November 2022, citing various neural network methods, including LSTMs and Transformers, but not contextualized Flair embeddings. The authors cite the following core approaches: 1) classifiers, 2) statistical machine translation, 3) neural machine translation, 4) edit-based approaches, and 5) language models for low-source and unsupervised GEC.

2.1 Flair Embeddings

Flair embeddings (Akbik et al., 2018) are *contextualized embeddings* trained without explicit notions of words and contextualized by their surrounding text. As they were launched, the embeddings were evaluated on four classic sequence labeling tasks: Named Entity Recognition (NER)-English, NER-German, Chunking, and Part-of-Speech (POS)-tagging. Akbik et al. reported improved scores on several datasets. The embeddings are trained with a forward-backward Recurrent Neural Network (RNN), and can be stacked before being applied to a particular problem.

Flair embeddings are pre-trained on large unlabeled corpora, they capture word meaning in context, and they model words as sequences of characters, which helps them with modeling rare and misspelled words. Thus, applying them to a sequence labeling problem such as GED is an interesting research option. Akbik et al. (2019b) launched *pooled* contextual embeddings to address the shortcoming of dealing with rare words in underspecified context. The pooled embeddings aggregate contextualized embeddings as they are encountered in a dataset. The Flair embeddings are released for all of the languages studied in MultiGED-2023, as well as in a multilingual version, covering more than 300 languages.[2]

In addition to the authors' experiments, Flair embeddings have previously been applied to sequence labeling in the biomedical domain (Sharma and Jr., 2019; Akhtyamova and Cardiff, 2020), achieving similar performance to alternatives like BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019), despite being computationally cheaper. Santos et al. (2019) and Consoli et al. (2020) achieved state-of-the-art results on doing NER on Portuguese literature in the geoscience domain. Wiedemann et al. (2019) compared Flair embeddings to BERT in a word sense disambiguation task, and argued that the latter models were better able to find the right sense of polysemic words. Syed et al. (2022) combined Flair and BERT embeddings for concept compilation in the medical domain, reporting improved results with a hybrid artificial neural network model, which concatenates the two embedding types. The FlairNLP framework also offers this functionality.

3 Data and preprocessing

Six datasets in five languages were used for the MultiGED-2023 shared task, ranging from 8k to 35k sentences.[3] The data loaded unproblematically, with the exception of line 96487 in the Swedish training corpus, a UTF-8 character that broke scripts. Specifically, embeddings were created with wrong dimensions. This character was replaced by the string ‘FOO’ in the experiments on this corpus to work around this problem. Additionally, line 149 in the Swedish test corpus and line 5351 in the Italian test corpus caused some problems. Because the FlairNLP framework, in contrast to, for instance, OpenNMT (Klein et al., 2017), parses the vertical format directly, no other preprocessing steps were necessary.

For the English Realec corpus, only a development and a test file were provided. More details are provided by Volodina et al. (2023).

4 Method

The FlairNLP framework was used to conduct the experiments presented below. After the data was loaded, it was passed to a processing pipeline, which is a sequence-to-sequence labeler consisting of a bi-directional LSTM (long short-term memory; Hochreiter and Schmidhuber, 1997) with an optional Conditional random field (CRF; Lafferty et al., 2001) classifier on top. Next, the model uses the training and development corpora for training, as well as F_1 scoring.

The architecture of the models can be adapted, e.g., in terms of recurrent neural network (RNN) layers, RNN type (RNN, LSTM or GRU — gated recurrent unit), the number of hidden units and training epochs, and the optional use of CRF. Additionally, the Tensorboard[4] system was used to monitor training progress.

[2] https://github.com/flairNLP/flair/blob/master/resources/docs/embeddings/FLAIR_EMBEDDINGS.md

[3] https://github.com/spraakbanken/multiged-2023/

[4] Part of TensorFlow (Abadi et al., 2015).

FlairNLP can combine several corpora into a `MultiCorpus` object, which builds a *monolithic* model of several corpora. This object can be used to train and test a single model on a collection of corpora, analogously to how a `Corpus` object can be used to do training and inference of one corpus for same. In the following, such a monolithic MGED model is considered multilingual, in contrast to several smaller, *individual* models, one for each language or dataset. While it is possible to have different models for different languages and direct input by means of language identification prior to inference, this distinction is made for clarity in separating the approaches.

Since the Realec corpus only came with development and test files, it was used differently than the other corpora: the English language was covered by the monolithic models and the individual model for the English FCE corpus, so the Realec test corpus was tested on this model and submitted to CodaLab (Pavao et al., 2022) for evaluation. The Realec dev corpus was not used in training.

4.1 Exploring Embeddings vs. Architecture

As a Bi-LSTM-CRF model is sensitive to initialization, a wide range of RNN layers (2, 6, 12, 24), hidden units (128, 256, 512) were explored as well as using GRUs and standard LSTMs. While there is a scope for tweaking the results, none of these configurations resulted in markedly better performance, with the exception of models with very few layers that were unable to converge to anything but same-labeling the entire corpus. For the results reported in Section 5, the choice for RNN type was LSTM, and the number of layers was 10.

4.2 System Submitted to the Test Phase of the Shared Task

The system submitted to the test phase was a monolithic multilingual system, which used the multilingual Flair embeddings. The architecture was a Bi-LSTM-CRF sequence labeler with only one layer and using no CRF. While the system was able to learn for all languages simultaneously, the performance was weak, especially in terms of recall and $F_{0.5}$.

5 Experimental Results

The experiments presented below were all carried out with the RNN type LSTM, using 10 layers with 256 hidden units, no use of CRF, and with a tag dictionary of only $[c, i]$. The experiments consisted of two stages: initially, five systems (including only one English model) were developed for each language using both Flair and BERT embeddings; subsequently, two monolithic models were created employing cased multilingual Flair and BERT embeddings. After presenting the scores of the simple system submitted to the shared task, these two types of experiments will be presented.

Table 1: Monolithic system submitted to the test phase of the shared task.

Dataset	Precision	Recall	$F_{0.5}$
Czech	80.65	6.49	24.54
English (FCE)	81.37	1.84	8.45
English (Realec)	51.34	1.13	5.19
German	83.56	15.58	44.61
Italian	93.38	19.84	53.62
Swedish	80.12	5.09	20.31

5.1 System Submitted to the Test Phase of the Shared Task

Table 1 shows the results of the system that was submitted to the test phase of the shared task, which was discussed above. Using only one RNN, layer, the monolithic model using Flair embeddings did get good precision on some datasets, but at the cost of recall and $F_{0.5}$ score. Only the score on the Italian dataset came close to the models using 10 layers in $F_{0.5}$ terms.

5.2 Individual Models for each Language

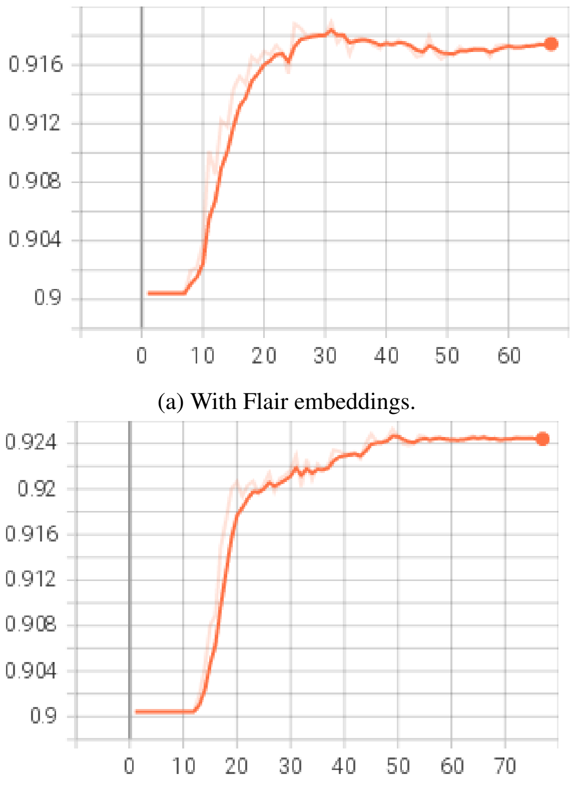
Figure 2 shows how the English FCE model (as an example) developed toward convergence and Table 2 exhibits the results in tabular form. The FCE models were chosen randomly as two samples of the ten models that were built in total. The results are better for BERT embeddings across all languages, and the differences are the largest for the smaller datasets, Swedish and Italian, than the larger English, German, and Czech, which is highlighted in the extra column of Table 2b.

BERT models are available for these languages in the Huggingface[5] interface: Czech (Sido et al., 2021), English (Devlin et al., 2019), German[6], Italian[7], and Swedish (Malmsten et al., 2020).

[5] https://huggingface.co/

[6] https://www.deepset.ai/german-bert

[7] https://huggingface.co/dbmdz/bert-base-italian-cased



(a) With Flair embeddings.

(b) With BERT embeddings.

Figure 2: Development corpus score per epoch until convergence for the English FCE model.

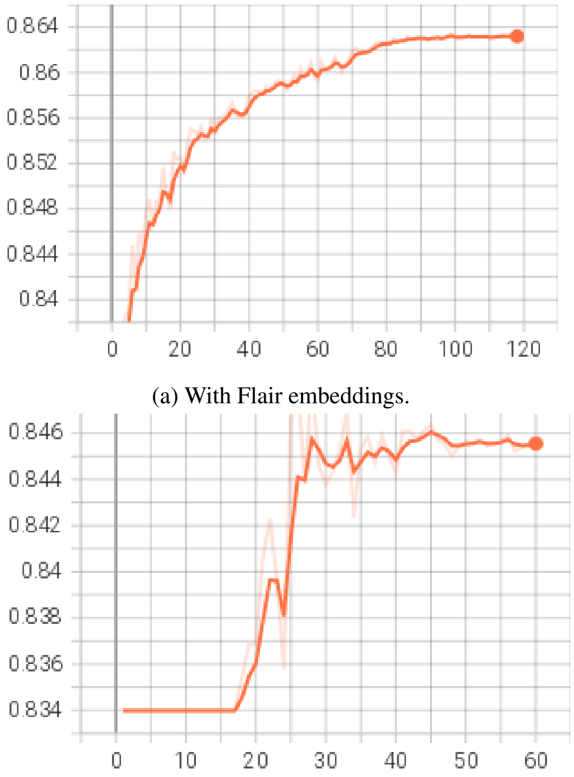
Table 2: Comparison of individual models. The 'Diff' column shows the difference between the two models (Flair vs. BERT). The biggest difference in **bold**, the smallest in *italics*.

(a) Individual models built with Flair embeddings.

Dataset	Prec.	Rec.	$F_{0.5}$
Czech	75.3	39.46	63.73
En (FCE)	65.49	33.01	54.72
En (Realec)	41.52	28.12	37.91
German	78.06	56.37	72.48
Italian	70.29	27.28	53.44
Swedish	57.44	26.85	46.78

(b) Individual models built with BERT embeddings.

Dataset	Prec.	Rec.	$F_{0.5}$	Diff
Czech	80.2	47.22	70.37	6.64
En (FCE)	71.13	41.5	62.25	7.53
En (Realec)	44.9	35.2	42.56	*4.65*
German	81.99	65.48	78.05	5.57
Italian	83.45	63.54	78.53	25.09
Swedish	80.64	60.1	75.48	**27.7**



(a) With Flair embeddings.

(b) With BERT embeddings.

Figure 3: Development corpus score per epoch until convergence for the monolithic models.

Table 3: Comparison of monolithic models. The 'Diff' column shows the difference between the two models (Flair vs. BERT). The biggest difference in **bold**, the smallest in *italics*.

(a) Monolithic model built with Flair embeddings.

Dataset	Prec.	Rec.	$F_{0.5}$
Czech	70.21	21.05	47.85
En (FCE)	66.76	10.13	31.52
En (Realec)	41.91	9.23	24.54
German	72.35	33.2	58.54
Italian	84.02	28.89	60.81
Swedish	67.57	19.45	45.2

(b) Monolithic model built with BERT embeddings.

Dataset	Prec.	Rec.	$F_{0.5}$	Diff
Czech	54.07	20.43	40.68	-7.17
En (FCE)	68.51	41.04	60.42	**28.9**
En (Realec)	42.07	35.1	40.46	15.92
German	59.6	26.55	47.72	-10.82
Italian	47.55	20.78	37.8	*-23.0*
Swedish	50.04	24.36	41.32	-3.88

5.3 Monolithic Models for all Languages

Figure 3 shows how the monolithic model developed towards convergence for both embedding types, and Table 3 exhibits the results in tabular form. The multilingual and cased BERT model and the corresponding Flair model were used for the embeddings. The results are markedly better for the English datasets but worse for the others, in particular Italian.

6 Discussion

As expected, the Flair embeddings performed worse than the more expensive BERT models individually. The results show that the Flair embeddings were performing closer to the BERT models for the larger corpora, with a larger difference for the smaller Italian and Swedish corpora. The masked language model training of BERT could introduce more imbalances when the corpora have different sizes. Possibly, the Flair embeddings need more training data to perform well.

It was a more mixed picture for the monolithic MGED models, where the BERT embeddings scored better for the English but worse for the other languages. Unlike for the individual models, performance was actually worse than with Flair embeddings, the reasons for which should be further explored.

In some experiments, the training process would get stuck in local minima, which converged to models that categorized all words as *c*. Anecdotally, fewer experiments were necessary to make the experiments using Flair embeddings to converge to a result other than a one-category (thus, meaningless) result. In contrast, the monolithic models using BERT embeddings were harder to get to converge to a result with both correct and incorrect predictions. Thus, several experiments were necessary to get a meaningful result out, although those models were performing better.

Furthermore, some experiments on model architecture were conducted by changing the RNN type, number of layers, or the dimensionality of the hidden state vector. While no notable differences in results were discovered in this exploratory phase, a potential for tweaking the models to increase performance on the test set likely remains.

As a consequence of an implementation error, the results submitted to the test phase of MultiGED-2023 were revised and turned out to be better. The errors were due to the FlairNLP system outputting a labeling of the test set, which was different from using the best model from training on the dataset, which caused minor differences in scoring. However, the substantial performance gain in the results presented above compared to the results submitted to the test phase stems from the architectural change to the system, whereby more RNN layers were added. The submitted system was simple, as the exploratory phase of getting the setup to produce results reliably had just been completed. As the scoring in CodaLab was (and is) available in the open phase, more work could be done, both in development and comparison terms.

For monolithic models, the multilingual BERT models are resource-demanding. Since the experiments were carried out on a multiuser HPC (high-performance computing) grid with many outside factors influencing performance, training times cannot be compared directly. Approximately and informally, however, the monolithic jobs with BERT embeddings could take 36 hours to converge, while the corresponding jobs with Flair embeddings converged in 6–8 hours.

7 Conclusion and Future Work

The research questions posed concerned (i) the feasibility of using Flair embeddings on an MGED task and (ii) monolithic vs. individual models.

The Flair embeddings were definitely feasible. For the larger datasets, performance neared BERT models, and did better on non-English languages for the monolithic approach. The monolithic approach did, however, perform worse than the individual models for both Flair and BERT embeddings. Thus, more research is needed to improve the monolithic approaches, with the gap in performance in the presented results too big to ignore.

For future work, hybrid solutions could be explored, where Flair and BERT embeddings are stacked. There is also room for further exploring the parameter space of the sequence-to-sequence labeling architecture, as well as leveraging newer and larger language models for embeddings. In addition, it would be interesting to apply $F_{0.5}$ scoring in training, as opposed to the default F_1 scoring in the FlairNLP framework that was used in the experiments reported here.

Acknowledgments

The computations were performed on resources provided by Sigma2 — the National Infrastructure for High Performance Computing and Data Storage in Norway.[8]

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin andand Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019b. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Liliya Akhtyamova and John Cardiff. 2020. LM-based word embeddings improve biomedical named entity recognition: A detailed analysis. In *Bioinformatics and Biomedical Engineering*, pages 624–635, Cham. Springer International Publishing.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *CoRR*, abs/2211.05166.

Bernardo Consoli, Joaquim Santos, Diogo Gomes, Fabio Cordeiro, Renata Vieira, and Viviane Moreira. 2020. Embeddings for named entity recognition in geoscience Portuguese literature. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4625–4630, Marseille, France. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of Sweden — making a Swedish BERT. *CoRR*, abs/2007.01658.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. CodaLab competitions: An open source platform to organize scientific challenges. Technical report, Université Paris-Saclay, Paris, France.

Joaquim Santos, Bernardo Consoli, Cicero dos Santos, Juliano Terra, Sandra Collonini, and Renata Vieira. 2019. Assessing the impact of contextual embeddings for Portuguese named entity recognition. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 437–442, Salvador, Brazil. IEEE.

Shreyas Sharma and Ron Daniel Jr. 2019. BioFLAIR: Pretrained pooled contextualized embeddings for biomedical sequence labeling tasks. *CoRR*, abs/1908.05760.

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. 2021. Czert — Czech BERT-like model for language representation. *CoRR*, abs/2103.13031.

Shorabuddin Syed, Adam Jackson Angel, Hafsa Bareen Syeda, Carole France Jennings, Joseph VanScoy, Mahanazuddin Syed, Melody Greer, Sudeepa Bhattacharyya, Meredith Zozus, Benjamin

[8] http://www.sigma2.no

Tharian, and Fred Prior. 2022. The h-ANN model: Comprehensive colonoscopy concept compilation using combined contextual embeddings. In *Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies — HEALTHINF*, pages 189–200, Virtual. INSTICC, SciTePress.

Elena Volodina, Christopher Bryant, Andrew Caines, Orphée De Clercq, Jennifer-Carmen Frey, Elizaveta Ershova, Alexandr Rosen, and Olga Vinogradova. 2023. MultiGED-2023 shared task at NLP4CALL: Multilingual grammatical error detection. In *Proceedings of the 12th workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL)*, Tórshavn, Faroe Islands.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS*, Erlangen, Germany.