

Boosting Language Models for Real-Word Error Detection

Corina Masanti^{1,2}^a, Hans-Friedrich Witschel²^b and Kaspar Riesen¹^c

¹*Institute of Computer Science, University of Bern, 3012 Bern, Switzerland*

²*Institute for Informations Systems, University of Appl. Sci. and Arts Northwestern Switzerland, 4600 Olten, Switzerland*
{corina.masanti, kaspar.riesen}@unibe.ch, hans-friedrich.witschel@fhmw.ch

Keywords: Boosting Techniques, Language Models, Synthetic Data, Real-Word Errors.

Abstract: With the introduction of transformer-based language models, research in error detection in text documents has significantly advanced. However, some significant research challenges remain. In the present paper, we aim to address the specific challenge of detecting real-word errors, i.e., words that are syntactically correct but semantically incorrect given the sentence context. In particular, we research three categories of frequent real-word errors in German, viz. verb conjugation errors, case errors, and capitalization errors. To address the scarcity of training data, especially for languages other than English, we propose to systematically incorporate synthetic data into the training process. To this end, we employ ensemble learning methods for language models. In particular, we propose to adapt the boosting technique to language model learning. Our experimental evaluation reveals that incorporating synthetic data in a non-systematic way enhances recall but lowers precision. In contrast, the proposed boosting approach improves the recall of the language model while maintaining its high precision.

1 INTRODUCTION

The challenge of automatic error detection and correction in text has been a persistent problem in pattern recognition for document analysis. The development of advanced technological tools, notably transformer-based models (Vaswani et al., 2017), has facilitated the emergence of powerful proofreading systems. Yet, accurately detecting and correcting *real-word errors* remains a significant challenge for current proofreading systems. Real-word errors refer to words in texts that exist in the underlying dictionary but are incorrect in the context of the sentence. One open issue in detecting real-word errors is that there is limited data available for training the models, especially for languages other than the dominant language in research (namely English).


To counteract this limitation, we propose to incorporate synthetic data in the training process for three categories of real-word errors frequently encountered in German text, viz. conjugation errors in verbs, wrong case selection, and capitalization errors.


The first contribution of this paper is that we generate high-quality synthetic data from a real-world


text data set provided by a Swiss proofreading agency that can be used for model training. In addition to the introduction of a novel and large-scale synthetic data set, the second major contribution of this paper is that we propose to incorporate ensemble learning methods for language models. Actually, a few approaches have been proposed that combine ensemble learning methods with language models. One such strategy, known as *boosted prompting*, is inspired by classical boosting algorithms. This method iteratively augments the prompt set with new prompts that better generalize regions of the target problem space where the previous prompts underperform (Pitis et al., 2023). Another approach is to train multiple models and combine them for the final output. In (Li et al., 2019), CNN-based and transformer-based models were combined to tackle the challenge of grammatical error correction.

In the present paper, we propose to employ boosting techniques to enhance the training process of language models and ultimately improve the accuracy of language models for detecting real-word errors. To the best of our knowledge, this is the first time that boosting is used in combination with language models in this specific way and for this particular task.

The remainder of this paper is structured as follows. Section 2 reviews the related work. In Sec-

^a <https://orcid.org/0009-0002-4104-6315>

^b <https://orcid.org/0000-0002-8608-9039>

^c <https://orcid.org/0000-0002-9145-3157>

tion 3, we describe the data set used, and in Section 4, we outline the proposed method for boosting the language model. Section 5 presents and discusses the results obtained with the model, and Section 6 summarizes our findings and suggests directions for future research.

2 RELATED WORK

Detection and correction of errors in text documents is a widely researched topic in *document analysis* and *natural language processing* (Bryant et al., 2023). Depending on the type of errors, various approaches exist. This section provides a brief overview of the current state of the art in this field.

For non-word errors, dictionary-based methods are effective in identifying incorrect words. However, when it comes to real-word errors or the presence of non-dictionary words, such as uncommon proper nouns (e.g., product names), rare words, or foreign language terms, these approaches are inadequate (Hládek et al., 2020). Typically, contextual models are used to address these more complex error categories (Pirinen and Lindén, 2014). A common approach involves using language models that estimate the likelihood of a word’s occurrence based on its surrounding context. For example, trigram models are often utilized for this purpose (Wilcox-O’Hearn et al., 2008).

With the introduction of the transformer architecture (Vaswani et al., 2017), more capable models like BERT or GPT became popular and enabled more advanced context modelling. Many recent techniques address spell checking and correction simultaneously (e.g., (Moslem et al., 2023)). One significant challenge involves preventing the language models from excessively editing incorrect sentences, as this might alter their meaning (Coyne et al., 2023). Another issue of sophisticated methods for both automatic error detection and correction is that they rely on large training data. More precisely, these models require a substantial amount of data containing pairs of sentences, each with an erroneous version and its corresponding correction (Tan et al., 2020). Over the past decade, several data sets have become well-known and frequently utilized for training and assessing error detection and correction systems. These data sets are often presented as part of shared tasks (Bryant et al., 2023). Well-known data sets are, for example, the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) or the Cambridge English Write & Improve (W&I) and LOCNESS corpus (Bryant et al., 2019) for the BEA 2019 task.

However, the data sets for training models for error detection and correction are heavily skewed towards English. That is, research results and data sets in English are available, while those for other languages are notably limited (Etoori et al., 2018). There are some efforts to address this limitation. For instance, the Multilingual Grammatical Error Detection (MultiGED) shared task (Volodina et al., 2023), introduced in 2023, aims to address the imbalance in language representation. In addition to English data sets, MultiGED incorporates languages typically underrepresented, such as Czech, German, Italian, and Swedish.

A potential solution to the scarcity of data for underrepresented languages involves incorporating synthetic training data. There are multiple approaches to generate synthetic data, most of them can be categorized as *noise injection* or *back-translation* (Kiyono et al., 2019). For noise injection, one directly injects noise into a correct sentence. One approach of this category is to extract correction patterns from public data sets and apply the inverse of these corrections to error-free sentences to simulate human behaviour (Yuan and Felice, 2013). When using back-translation, one trains a reverse model that generates an ungrammatical sentence from a given grammatical sentence (e.g., (Rei et al., 2017)).

3 DATA SET AND TASK

Recently, a new data set of text documents from a Swiss proofreading company has been introduced (Masanti et al., 2023). This data set includes documents from clients from various industries, including pharmaceuticals, banking, insurance, retail, communications, and more. It also covers a wide range of document types, such as annual reports, letters, technical documentation, legal texts, advertising materials, presentation slides, social media content, newsletters, websites, magazine articles, and others. Reflecting the multilingual setting of Switzerland, many documents include translations relevant to each Swiss region, resulting in texts in German, French, Italian, and English, with German being the predominant language.

The data set is a comprehensive and unique collection of approximately 50,000 documents, with sentences manually annotated by proofreading experts. This data set is particularly well-suited for both our novel method and the subsequent experiments due to the presence of complex errors, such as real-word errors.

In this paper, we focus on detecting real-word

Table 1: Sample sentences from the data set illustrating each category of the real-word errors with their corresponding corrections. The corrections are emphasized in boldface.

Category	Original Sentence (S) and its Correction (C)
Case	<p>S: [...] nach der Durchführung eines offenen, zweistufigen Dialogverfahren [...]</p> <p>C: [...] nach der Durchführung eines offenen, zweistufigen Dialogverfahrens [...]</p>
Verb	<p>S: Falls du dir Sorgen und Gedanken macht, [...]</p> <p>C: Falls du dir Sorgen und Gedanken machst, [...]</p>
Capitalization	<p>S: [...] auf der Sie ihre Daten eingeben müssen, [...]</p> <p>C: [...] auf der Sie Ihre Daten eingeben müssen, [...]</p>

errors in German. German’s linguistic complexity makes it an ideal language for studying real-word errors. To this end, we filter errors where the erroneous word in the original sentence can be found in a German dictionary. From this, we categorize the sentences into three categories of real-word errors, namely *case errors*, *verb errors*, and *capitalization errors*. We count an error as a *case error* if the original word is not in the correct case (nominative, genitive, dative, accusative), including mistakes where a word should be written in singular instead of plural and vice versa. A *verb error* occurs when the concerning word is a verb that is corrected in conjugation or tense. The third category involves the capitalization of words. A *capitalization error* in this context is a word incorrectly written in lower or upper case. In the German language, this can happen in various ways. For instance, words that are not nouns but are used as nouns in the context of a sentence need to be capitalized.

Table 1 shows one example sentence per category. The data set for case errors includes 2,920, the one for verb errors contains 856 samples, and the capitalization data set holds 3,352 samples.

Since the actual data sets are too small to effectively train error-detection models, we propose to generate synthetic data from the real-world data set for our novel method as follows. For each category, we extract the error patterns of the data sets (similar to the method proposed in (Yuan and Felice, 2013)). More precisely, we extract word pairs (*incorrect-word*, *correct-word*) that correspond to a word before the correction and the version of this word after correction, respectively. For example, the capitalization error shown in Table 1 would produce the error pair (‘ihre’, ‘Ihre’). Using these pairs, we can inject errors by inverting the correction, meaning that we search for the correct word in the error pair

Table 2: Examples of synthetic data generation per category. The injected error is visualized in boldface.

Category	Original Correct Sentence (from Wikipedia)	Sentence with Injected Error
Case	Edgardo Massa gewann im Doppel zwei Titel.	Edgardo Massa gewann in Doppel zwei Titel.
Verb	Ein konsequentes Raster bilden die Fenster des Wernerwerk-Hochhauses in Berlin.	Ein konsequentes Raster bildet die Fenster des Wernerwerk-Hochhauses in Berlin.
Capitalization	Mit dem Ehemann Antanas hat sie den Sohn Vaidotas und die Tochter Asta.	Mit dem Ehemann Antanas hat Sie den Sohn Vaidotas und die Tochter Asta.

(e.g., ‘Ihre’) in an additional data set containing grammatically correct sentences and swap this word with the incorrect word from the pair (e.g., ‘ihre’). We use Wikipedia text data extracted with WikiExtractor (Attardi, 2015) for the definition of a large set of correct samples. Table 2 shows one example sentence from Wikipedia and the corresponding error injection for each error category.

Using this method, we create a large set of synthetic data with about 31.8 million sentences in total. Since using the complete set of synthetic data for training would be computationally too expensive, we select 50,000 samples from the complete set of sentences to create a baseline data set for each error category while reserving the remaining 31.7 million samples for strategic data augmentation by means of boosting (detailed in Section 4). In addition to the three error categories case, verb, and capitalization errors, we apply this procedure to a combined set that includes all three error types.

In our evaluation, we balance the data sets so that half of the sentences contain an error of the specified category and the other half is error-free (the corresponding sentences without errors). Thus, we generate a data set of 100,000 samples for each error category and each sentence in the three data sets is marked as correct (0) or incorrect (1).

To align with the distribution of the real-world data set, we ensure that the occurrence of an error pair reflects the actual error frequency. For example, if the error pair (‘ihre’, ‘Ihre’) occurs in 2% of the erroneous sentences in the original real-world data set, we aim to introduce a similar proportion in the synthetic data set. However, as some words, especially Swiss-specific terms may be rare or absent in Wikipedia data, we were only able to apply this heuristic approximately.

The complete process is visualized in Figure 1. In

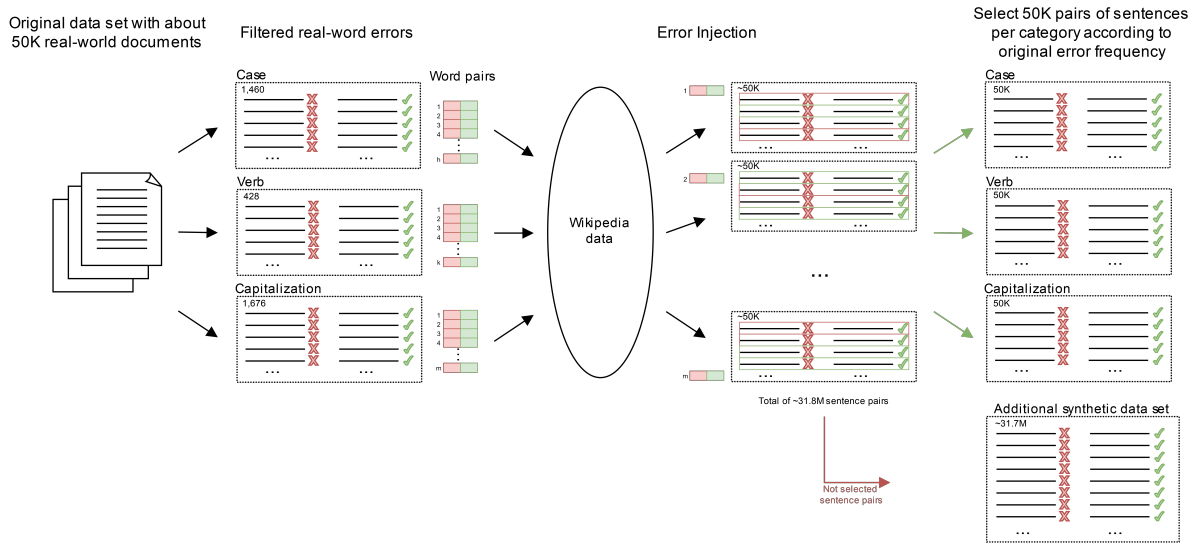


Figure 1: Visualization of the complete synthetic data generation process.

summary, we begin with the real-world data set of approximately 50,000 documents. From this data set, we extract sentences containing real-word errors and categorize them into case, verb, and capitalization errors. The numbers in the figure represent sentence pairs, consisting of the incorrect sentence and its corresponding correction. Therefore, the final data sets contain twice as many individual sentences as displayed in the figure. We extract the word pairs from the three categories. Using these word pairs as error patterns, we perform error injection using Wikipedia data as the source of correct sentences.

Since we balance the data sets to include 50% correct sentences and 50% incorrect sentences, this is a binary classification task. This means that this data set can be used to develop models that can reliably assign a label to a sentence indicating whether it contains an error (label=1) or not (label=0).

Although synthetic data may not fully capture the complexity and variability of natural language errors, we argue that our approach reduces this concern. By injecting errors from a real-world data set into Wikipedia data and aligning the error distribution with that of the real-world data set, the synthetic errors hold a strong connection to the original errors.

4 BOOSTING LANGUAGE MODELS

Our research focuses on the requirements of an error detection model intended to be used by professional proofreaders to facilitate their daily workload. False positives of a given model can easily be fixed by

professional proofreaders, while the risk of any error being overlooked should be minimized. Hence, high recall values are crucial to ensure that all errors are identified by a model while maintaining reasonable precision.

To fulfil these requirements, we propose in this paper to progressively add synthetic data from the synthetic data set mentioned in Section 3 to the training set and systematically improve the model. Rather than performing this incremental addition randomly, we propose to apply the boosting technique (Freund, 1990; Schapire, 1990). Our goal is to incorporate only synthetic samples that address the current limitations of the model.

The proposed method is formalized in Algorithm 1. First, we fine-tune the model on the training data (line 3 in Algorithm 1). Then, we evaluate the model on the validation set (line 4 in Algorithm 1). After this step, we identify sentences where the model missed an error – that is, we search for samples carrying the actual label 1 (indicating an error), but the model predicted label 0. We extract the error pairs consisting of the incorrect word and its corrected counterpart for these misclassified sentences. If the error-pair has not yet been used in this epoch (see line 6 in Algorithm 1), we add up to N synthetic samples for each error pair into the training set for the next iteration (depending on availability). These samples are balanced, with $N/2$ samples containing the incorrect word (label = 1) and $N/2$ samples containing the corrected word (label = 0). We repeat this process n times, and finally, one can evaluate the model on an independent test set (see line 10 in Algorithm 1).

Theoretically, both the number of epochs n and the number of synthetic samples N that are added in each

step can be defined arbitrarily. In our experiments, we set $n = 5$ and $N = 500$, as the time required for the fine-tuning procedure increases drastically with a large amount of training data and we observed a stabilization of accuracy after five epochs at the latest.

Algorithm 1: Step-wise integration of synthetic data for model fine-tuning using boosting.

```

1 Initialize added_errors  $\leftarrow \emptyset$ ;
2 for epoch = 1 to n do
3   Fine-tune model on training data;
4   Evaluate on validation set;
5   for each misclassified
      (incorrect-word, correct-word) pair do
6     if pair not in added_errors then
7       Select up to N synthetic samples
          for this pair;
8       Add samples to training set;
9       Add pair to added_errors;
10 Test model on test set;

```

A potential drawback of this method is the need to fine-tune the model multiple times, which can be time-consuming with large data sets and complex models. However, we reduce the time required for each round of fine-tuning by using a subset of the whole data set in the first epoch and incorporating only the samples that significantly enhance the model’s learning process. This approach is particularly beneficial for large data sets containing redundancy or noise as well as highly imbalanced data sets where underrepresented or challenging samples can be prioritized. Moreover, in scenarios involving both real-world and synthetic data, this approach allows for initial fine-tuning with real-world data followed by selective incorporation of synthetic data, thereby minimizing the risk of overfitting to artificial error patterns.

5 EXPERIMENTAL EVALUATION

For the experimental evaluation, we use a transformer-based model, namely mBERT, which is short for *Multilingual Bidirectional Encoder Representation from Transformers* (Devlin et al., 2019). This model is based on the encoder component of the transformer architecture (Vaswani et al., 2017). We set the model parameters to a learning rate of $1.1e-5$, a batch size of 64 and trained the model for 20 epochs.

Table 3: Results of the baseline, random selection, and boosting experiments. We show the Precision (P), Recall (R), and Accuracy (A) achieved on the test set for each individual category and for all categories combined. The highest scores for each category are shown in boldface.

Category	Baseline	Random Selection	Boosting
Case	P = 0.9143	P = 0.9200	P = 0.9306
	R = 0.9266	R = 0.9308	R = 0.9535
	A = 0.9199	A = 0.9249	A = 0.9412
Verb	P = 0.9457	P = 0.9248	P = 0.9418
	R = 0.9331	R = 0.9425	R = 0.9718
	A = 0.9398	A = 0.9330	A = 0.9559
Capitalization	P = 0.9413	P = 0.9133	P = 0.9500
	R = 0.9431	R = 0.9448	R = 0.9594
	A = 0.9421	A = 0.9276	A = 0.9545
Combined	P = 0.8897	P = 0.9029	P = 0.9012
	R = 0.8084	R = 0.9097	R = 0.9179
	A = 0.8541	A = 0.9056	A = 0.9086

As mentioned in Section 3, we focus on the task of detecting real-word errors derived from three categories. For each error-category, we conduct the following three experiments.

- Baseline:** We fine-tune the model with the synthetic data set where we split the data set into 50% for training, 10% for validation, and 40% for testing.
- Boosting:** We start with the same data set as the baseline experiment and use the method described in Section 4 to add synthetic data into the training set and boost the model.
- Random Selection:** We augment the training set of the baseline experiment with the same amount of samples used in the boosting procedure, but we select the samples randomly from the synthetic data set. This experiment serves as ablation study to confirm that the benefit of our model lies in the boosting technique rather than the pure addition of data to the training set.

Figure 2 shows the validation results of the boosting process with $n = 5$ epochs (the subfigures show the results of the evaluation step on the validation set for all three error categories). Overall, accuracy, precision, and recall increase with each epoch, indicating consistent improvement of the model’s predictions. The first time systematically adding synthetic data makes the most difference, particularly pronounced in the category of case errors.

Table 3 summarizes the final results on the test set of the three experiments mentioned above. Across all error categories, the boosting technique produces

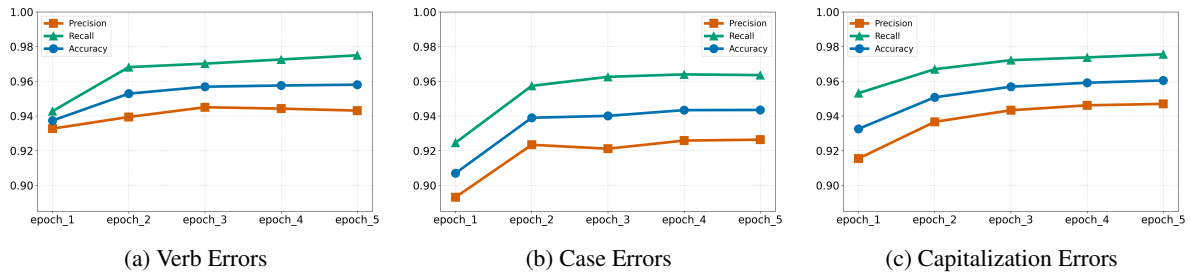


Figure 2: Results of boosting technique for each individual error category. The results show the Precision, Recall, and Accuracy scores achieved on the validation set for each epoch.

the highest recall and accuracy value while maintaining high precision. We observe the most drastic improvement for verb errors, with an increase in recall of around 4 percentage points. We even observe the highest precision value with the boosting technique for case and capitalization errors. For verb errors, the baseline experiment holds the highest precision value with only a difference of approximately 0.4% to the boosting technique.

Regarding the ablation study for verb and capitalization errors, we can report that random selection improves recall but results in lower precision, particularly for capitalization errors. These results indicate that boosting is clearly beneficial. That is, systematically augmenting training sets with relevant synthetic data helps the model to detect more errors without hurting precision.

The classification task becomes more challenging when all error types are combined (category combined). Here we observe an accuracy of around 0.85 compared to the other categories with accuracies between 0.92 and 0.94. For the combined category, we observe that the boosting technique produces higher precision, recall, and accuracy than the baseline experiment. However, precision is highest for random selection, with a difference of only about 0.2% to the boosting technique. The closer values between random selection and boosting may stem from a higher overlap of added synthetic samples incorporated in both techniques. Due to an increase in false negatives, we add more synthetic data. For the combined data set, there is an overlap in error patterns of 22% whereas for the individual categories of case, verb, and capitalization errors, the overlap is only 8%, 5%, and 11%, respectively.

6 CONCLUSIONS AND FUTURE WORK

Despite powerful transformer-based language models, some significant research challenges remain, par-

ticularly the specific challenge of detecting real-word errors in documents. In this paper, we propose a novel method for providing a language model with synthetic data to improve model performance. With the proposed approach, we target areas where a model is weak and has difficulty classifying samples correctly and address challenges and limitations directly. The results show that non-systematic addition of synthetic data increases recall, but precision can be affected. With the proposed technique of boosting, on the other hand, recall improves while precision remains high. We therefore conclude that boosting language models can work significantly better than randomly integrating large amounts of additional data. In a detailed investigation, we found that the boosting technique improves performance in each epoch. However, the increase in performance flattens out with an increasing number of epochs and thus, a larger number of epochs would not provide sufficient benefit to justify the additional computational cost.

For future work, we propose adapting the noise injection strategy to better reflect the real-world data set. For example, when a capitalization error occurs after a colon, we would specifically search for the correct word in the same context – after a colon in this case – rather than simply replacing the word in every occurrence. In addition, it would be interesting to assess the effectiveness of our approach when applied to models larger than mBERT. Furthermore, it would be beneficial to evaluate the approach using a real-world data set independent of the one used for training. During this study, it was not feasible due to limited data availability of real-word errors in the German language. However, seeing a trend in the public availability of data sets beyond the predominant language English, we hope to compile a sufficiently large data set for these error types. Finally, it would be interesting to see this approach applied in other domains with similar challenges.

REFERENCES

- Attardi, G. (2015). Wikiextractor. <https://github.com/attardi/wikiextractor>.
- Bryant, C., Felice, M., Andersen, Ø. E., and Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. In Yannakoudakis, H., Kochmar, E., Leacock, C., Madnani, N., Pilán, I., and Zesch, T., editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, pages 52–75. Association for Computational Linguistics.
- Bryant, C., Yuan, Z., Qorib, M. R., Cao, H., Ng, H. T., and Briscoe, T. (2023). Grammatical error correction: A survey of the state of the art. *Comput. Linguistics*, 49(3):643–701.
- Coyne, S., Sakaguchi, K., Galvan-Sosa, D., Zock, M., and Inui, K. (2023). Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction.
- Dahlmeier, D., Ng, H. T., and Wu, S. M. (2013). Building a large annotated corpus of learner english: The NUS corpus of learner english. In Tetreault, J. R., Burstein, J., and Leacock, C., editors, *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, June 13, 2013, Atlanta, Georgia, USA*, pages 22–31. The Association for Computer Linguistics.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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 4171–4186. Association for Computational Linguistics.
- Etoori, P., Chinnakotla, M., and Mamidi, R. (2018). Automatic spelling correction for resource-scarce languages using deep learning. In Shwartz, V., Tabasum, J., Voigt, R., Che, W., de Marneffe, M., and Nissim, M., editors, *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, Student Research Workshop*, pages 146–152. Association for Computational Linguistics.
- Freund, Y. (1990). Boosting a weak learning algorithm by majority. In Fulk, M. A. and Case, J., editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT 1990, University of Rochester, Rochester, NY, USA, August 6-8, 1990*, pages 202–216. Morgan Kaufmann.
- Hládek, D., Staš, J., and Pleva, M. (2020). Survey of automatic spelling correction. *Electronics*, 9(10):1670.
- Kiyono, S., Suzuki, J., Mita, M., Mizumoto, T., and Inui, K. (2019). An empirical study of incorporating pseudo data into grammatical error correction. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1236–1242. Association for Computational Linguistics.
- Li, R., Wang, C., Zha, Y., Yu, Y., Guo, S., Wang, Q., Liu, Y., and Lin, H. (2019). The LAIX systems in the BEA-2019 GEC shared task. In Yannakoudakis, H., Kochmar, E., Leacock, C., Madnani, N., Pilán, I., and Zesch, T., editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, pages 159–167. Association for Computational Linguistics.
- Masanti, C., Witschel, H.-F., and Riesen, K. (2023). Novel benchmark data set for automatic error detection and correction. In *International Conference on Applications of Natural Language to Information Systems*, pages 511–521. Springer.
- Moslem, Y., Haque, R., Kelleher, J. D., and Way, A. (2023). Adaptive machine translation with large language models. In Nurminen, M., Brenner, J., Koponen, M., Latomaa, S., Mikhailov, M., Schierl, F., Ransinghe, T., Vanmassenhove, E., Vidal, S. A., Aranberri, N., Nunziatini, M., Escartín, C. P., Forcada, M. L., Popovic, M., Scarton, C., and Moniz, H., editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 227–237. European Association for Machine Translation.
- Pirinen, T. A. and Lindén, K. (2014). State-of-the-art in weighted finite-state spell-checking. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing - 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II*, volume 8404 of *Lecture Notes in Computer Science*, pages 519–532. Springer.
- Pitis, S., Zhang, M. R., Wang, A., and Ba, J. (2023). Boosted prompt ensembles for large language models. *CoRR*, abs/2304.05970.
- Rei, M., Felice, M., Yuan, Z., and Briscoe, T. (2017). Artificial error generation with machine translation and syntactic patterns. In Tetreault, J. R., Burstein, J., Leacock, C., and Yannakoudakis, H., editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 287–292. Association for Computational Linguistics.
- Schapire, R. E. (1990). The strength of weak learnability. *Mach. Learn.*, 5:197–227.
- Tan, M., Chen, D., Li, Z., and Wang, P. (2020). Spelling error correction with bert based on character-phonetic. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1146–1150. IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*:

Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

- Volodina, E., Bryant, C., Caines, A., De Clercq, O., Frey, J.-C., Ershova, E., Rosen, A., and Vinogradova, O. (2023). Multiged-2023 shared task at nlp4call: Multilingual grammatical error detection. In *Proceedings of the 12th Workshop on NLP for Computer Assisted Language Learning*, pages 1–16.
- Wilcox-O’Hearn, L. A., Hirst, G., and Budanitsky, A. (2008). Real-word spelling correction with trigrams: A reconsideration of the mays, damerau, and mercer model. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 9th International Conference, CICLing 2008, Haifa, Israel, February 17-23, 2008, Proceedings*, volume 4919 of *Lecture Notes in Computer Science*, pages 605–616. Springer.
- Yuan, Z. and Felice, M. (2013). Constrained grammatical error correction using statistical machine translation. In Ng, H. T., Tetreault, J. R., Wu, S. M., Wu, Y., and Hadiwinoto, C., editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 52–61. ACL.



SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS