Exploring the Capacity of a Large-scale Masked Language Model to Recognize Grammatical Errors

Ryo Nagata
Konan University / Japan
nagata-acl2022@ ml.hyogo-u.ac.jp.

Manabu Kimura
GRAS Group, Inc. / Japan
manabu.kimura@gras-group.co.jp

Kazuaki Hanawa
RIKEN / Japan
Tohoku University / Japan
kazuaki.hanawa@riken.jp

Abstract

In this paper, we explore the capacity of a language model-based method for grammatical error detection in detail. We first show that 5 to 10% of training data are enough for a BERT-based error detection method to achieve performance equivalent to what a non-language model-based method can achieve with the full training data; recall improves much faster with respect to training data size in the BERT-based method than in the non-language model method. This suggests that (i) the BERT-based method should have a good knowledge of the grammar required to recognize certain types of error and that (ii) it can transform the knowledge into error detection rules by fine-tuning with few training samples, which explains its high generalization ability in grammatical error detection. We further show with pseudo error data that it actually exhibits such nice properties in learning rules for recognizing various types of error. Finally, based on these findings, we discuss a cost-effective method for detecting grammatical errors with feedback comments explaining relevant grammatical rules to learners.

1 Introduction

Recent studies have shown that masked language models pre-trained on a large corpus (hereafter, simply language models) achieve tremendous improvements over a wide variety of natural language processing tasks with fine-tuning. These results suggest that they are also effective in recognizing erroneous words and phrases, the task known as grammatical error detection. There has been, however, much less work on this aspect of grammatical error detection than in other tasks. One can argue that since language models are trained on language data produced by native speakers of a language (specifically, English in this paper), they might not work well on the target language data produced by non-native speakers of that language. In other words, English language models do not know at all about grammatical errors made by non-native speakers. Even apart from grammatical errors, the target language is different from the canonical English, meaning that it contains unnatural phrases and characteristic language usages that native speakers do not normally use, as Nagata and Whittaker (2013) demonstrate. These differences might affect performance of language model-based methods in grammatical error detection.

Actually, researchers have reported on performance of language models on grammatical error detection and correction. Cheng and Duan (2020) and Kaneko and Komachi (2019) have shown that BERT (Devlin et al., 2019)-based methods improve grammatical error detection performance in Chinese and English, respectively. Kaneko et al. (2020) and Didenko and Shaptala (2019) have shown a similar tendency in grammatical error correction. While these studies empirically prove the effectiveness of language models in grammatical error detection and correction, the questions of why and where language models benefit error detection/correction methods are left unanswered.

In this paper, we explore this aspect of language models in grammatical error detection to better answer the research questions. We first show that a BERT-based method incredibly quickly learns to recognize various types of error as summarized in Figure 1; it achieves only with 5 to 10% of training data an $F_{1.0}$ that a non-language model-based method can achieve with the full training data (the details will be described in Sect. 4). This implies that the BERT-based method (i) should have a good knowledge of the grammar required to recognize certain types of error and (ii) can transform it into error detection rules by fine-tuning with very few

Findings of the Association for Computational Linguistics: ACL 2022, pages 4107 - 4118

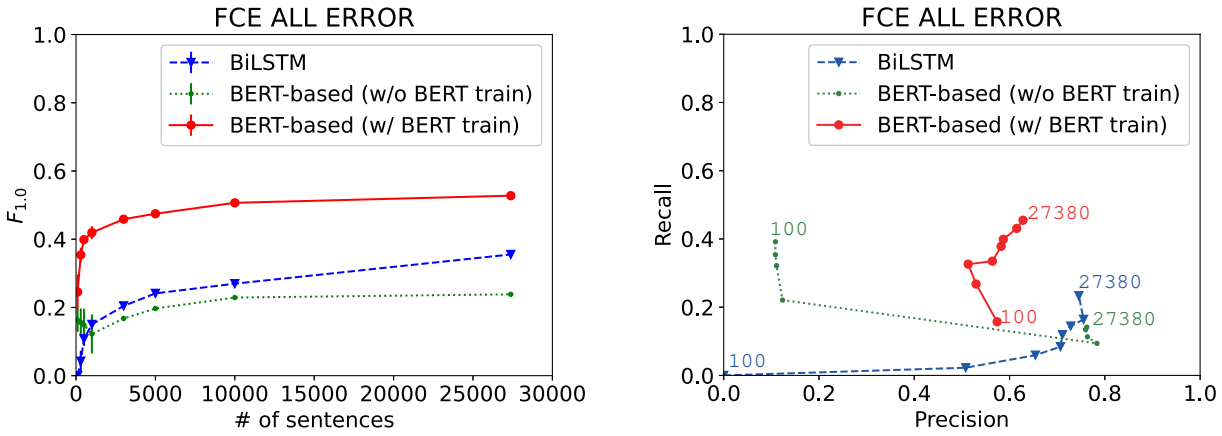

Figure 1: Detection Performance in Relation with Training Size (FCE).

training samples. Following this, we further show with real and pseudo error data why and where it gains in error detection, revealing the insights of language model-based methods. For instance, we show that the BERT-based method trained on few instances of a transitive verb with a preposition (e.g., **discuss about*) can detect the same type of error in other verbs (e.g., **approach to* and **attend in*). Finally, based on these findings, we discuss a cost-effective method for detecting grammatical errors with feedback comments explaining relevant grammatical rules to learners.

2 Related Work

Rei (2017) shows it is useful for neural error detection models to introduce a secondary language model objective together with the main error detection objective. Rei and Yannakoudakis (2017) compare several other auxiliary training objectives including Part-Of-Speech (POS) tagging and error type identification and find that the language model objective is the most effective. This line of work suggests that grammatical error detection benefits from language modeling, although these studies use BiLSTM-based language models instead of masked language models trained on a large corpus.

As mentioned in Sect. 1, several researchers have applied masked language models including BERT to grammatical error detection and correction. Cheng and Duan (2020) and Kaneko and Komachi (2019) show that error detection methods gain in recall and precision with the use of language models. Bell et al. (2019) use BERT-based contextual embeddings for grammatical error detection and compare it with other types of contextual embedding. They show the BERT-based contextual embeddings are effective in almost all error types provided by ERRANT (Bryant et al., 2017) although BERT is not fine-tuned in their study. Yuan et al. (2021) compare BERT, XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020) in grammatical error detection to show their effectiveness in grammatical error detection[1]. Kaneko et al. (2020) and Didenko and Shaptala (2019) also show performance improvements in grammatical error correction. To strengthen the findings of these previous studies, we will explore why and where error detection methods benefit from language models, revealing their generalization ability, in the following sections.

There has been a long history of studies that investigate the linguistic knowledge of language models including the work by Li et al. (2021); Ettinger (2020); Warstadt et al. (2020) to name a few. A popular approach is to test whether a language model assigns higher likelihood to the appropriate word than an inappropriate one, given context. The linguistic knowledge to be explored ranges from syntactic/semantic knowledge to common sense. These studies mostly use (i) synthetic test data: sentences that are generated synthetically by using a certain kind of template or (ii) perturbed test data: sentences that are generated by perturbing a natural corpus. Our work is different from these previous studies in two points: (i) to our best knowledge, we examine linguistic phenomena that have never been explored before in the conventional studies (e.g., subjects marked with a preposition and errors involving the usages of transitive and intransitive verbs); (ii) we use a real learner corpus with real

[1]ELECTRA is not a language model. It however contains an architecture similar to that of a language model.

errors as our test data.

Mita and Yanaka (2021) examine if an encoder-decoder neural network for grammatical error correction (not BERT-based) can learn the knowledge of grammar through the task of grammatical error correction. They target five error types: subject-verb agreement, verb form, word order, adjective/adverb comparison, noun number. They use both synthetic and real learner data. They report a negative answer to the research question except for word order errors. They also report that their model learns the knowledge to detect the target errors in their synthetic data. However, there is still room for debate in this argument because error positions tend to be rather obvious in their synthetic data (e.g., adjective forms erroneously used as adverbs almost always appear at the end of a sentence). Our study expands and deepens their findings for a wider variety of error types that are much more difficult to detect (in that it requires a much wider range of linguistic knowledge including POS, lexical, and syntactic knowledge).

3 Data and Methods

3.1 Real and Pseudo Error Data

In this paper, we use two kinds of data: real and pseudo data. Real data are error-annotated learner corpora while pseudo error data are automatically generated by perturbing a native English corpus.

For the real data, we use four English learner corpora: the First Certificate of English error detection dataset (FCE) (Yannakoudakis et al., 2011); NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013); BEA-2019 shared task dataset (BEA) (Bryant et al., 2019); the International Corpus Network of Asian Learners of English with feedback comments (ICNALE) (Nagata et al., 2020). We use the data splits provided by the creator except for ICNALE where we randomly split the essays into training, development, and test sets in the ratios of 85%, 7.5%, and 7.5%, respectively. Table 1 shows their statistics[2].

ICNALE provides information about errors in preposition use and their corresponding feedback comments. We use it to investigate in detail why and where language model-based methods gain an advantage. Their essay topics are controlled; they are written on either (a) *It is important for college students to have a part-time job.* or (b) *Smoking should be completely banned at all the restaurants in the country.*, which hereafter will be referred to as *PART-TIME JOB* and *SMOKING*, respectively. Each essay is manually annotated with errors in preposition use and their corresponding feedback comments. For example, the major errors in the corpus include deverbal prepositions (e.g., **include* → *including*), intransitive verbs with a direct object (e.g., **agree it* → *agree with it*), a verb phrase used as a noun phrase (**Learn English is difficult.* → *To learn/Learning English is difficult.*), and comparison between a phrase and a clause (e.g., **because an error* → *because of an error*); see the work (Nagata et al., 2020) for the details.

To investigate the relationship between the number of training sentences and detection performance, we randomly sample 100, 300, 500, 1,000, 3,000, 5,000, 10,000, and all sentences, resulting in eight sets of training data for each corpus[3]. Note that these training, development, and test sets con-

[2]The data development is still ongoing in the work (Nagata, 2019). For this work, we used data that had not been open to the public yet from the developer.

[3]In the sub-corpora A and C in BEA, only seven sets are used because they consist of less than 10,000 sentences.

Split	Training			Development			Test		
Statistics	sents	tokens	errors	sents	tokens	errors	sents	tokens	errors
FCE	27,380	435,768	41,277	2,129	33,720	3,335	2,581	40,498	4,374
ICNALE									
PART-TIME JOB	12,163	205,355	2,439	1,129	18,276	244	1,042	17,192	222
SMOKING	12,312	201,304	2,342	1,160	18,242	230	1,023	17,318	212
BEA									
A	9,244	160,818	24,520	1,014	17,417	2,566	1,014	18,106	2,801
B	11,410	207,252	20,580	1,261	22,435	2,261	1,261	22,806	2,362
C	9,410	179,156	8,649	1,020	19,035	990	1,020	20,392	1,052
NUCLE	16,969	433,787	38,723	2,120	54,799	4,019	2,120	56,804	4,406

Table 1: Statistics on Real Datasets.

tain error-free sentences.

For the pseudo error data, we use the 1998-2000 New York Times in the AQUAINT Corpus of English News Text (Graff, 2002) as a base corpus. We automatically generate erroneous sentences by injecting errors into them (one error per sentence). We first obtain chunks and parses by using spaCy[4]. Here, we only use sentences whose lengths are longer than three tokens and shorter than 26 to obtain reliable chunks and parses. We then add, remove, or replace a word in the sentences based on the analyses.

While we target all errors labeled as errors in the real data, we only target the following five error types in the pseudo error data:

Prepositional infinitive: *to*-infinitive with other prepositions than *to*.
(e.g., *a book to read* → **a book for read*)

Subject verb: Verb phrases used as a subject
(e.g., **Learn English is difficult.*)

Preposition + subject: Subjects used with a preposition
(e.g., **In the restaurant serves good food.*)

Transitive verb + preposition: Transitive verbs used with a preposition
(e.g., **We discussed about it.*)

Intransitive verb + object: Intransitive verbs taking a direct object
(e.g., **We agree it.*)

These five error types are selected with the following two criteria: (i) they are major errors in ICNALE; (ii) we can write a software program to generate pseudo errors based on chunks and parses. For example, we can find the subject of a sentence from its parse and then can add a randomly-chosen preposition before the subject noun phrase (e.g., *The restaurant serves good food.* → **In the restaurant serves good food.*). We randomly choose one of the following five prepositions: *at*, *about*, *to*, *in*, and *with* for addition and replacement; an exception is that we only use *for* for "Prepositional infinitive" (e.g., *a book to read* → **a book for read*), which often appears in ICNALE. Similarly, we can extract pairs of a verb and its direct object from parses and then can add one of the prepositions before the direct object noun phrase as in *discuss the matter* → **discuss about the matter*. We select the following transitive verbs as our targets: in training/development data: *answer*, *attend*, *discuss*, *inhabit*, *mention*, *oppose*, and *resemble*; in test data: *approach*, *consider*, *enter*, *marry*, *obey*, *reach*, *visit*. Similarly, we select the following intransitive verbs: in training/development data: *agree*, *belong*, *disagree*, and *relate*; in test data: *apply*, *graduate*, *listen*, *specialize*, *worry*. It should be emphasized that there is no overlap of the target transitive/intransitive verbs in the training/development and test data.

From the resulting pseudo error data, we randomly sample $2^k (1 \leq k \leq 10)$ sentences for each error type, resulting in ten sets of training data (e.g., when $k = 1$, the set comprises two instances of each error type, ten instances in total). We use these training sets to estimate the relationship between the number of training sentences and detection performance. For a validation set, we randomly sample 200 sentences for each error type. Similarly, we use a test set consisting of 200 sentences randomly sampled for each error type plus another 200 error-free sentences. The validation and test sets are fixed regardless of the training data.

3.2 Grammatical Error Detection Methods

This subsection describes the three methods to be explored and compared. Before looking into them, let us define grammatical error detection formally. Grammar error detection can be solved as a token classification problem[5]. We will denote a sequence of words and its length by $w_1, \ldots, w_i \ldots, w_N$ and N, respectively. We will denote the corresponding sequence of labels by $l_1, \ldots, l_i, \ldots, l_N$ where l_i corresponds to the label of w_i. We assume two sets of labels: (i) either C or E denoting *correct* or *erroneous* in the real data, respectively; and (ii) K labels for K error types plus C for *correct* in the pseudo error data. Then, grammatical error detection is defined as a problem of predicting the optimal label sequence given $w_1, \ldots, w_i \ldots, w_N$.

We use neural networks to predict the optimal label sequence. In this paper, training is repeated five times with different (but fixed) random seeds. The reported performance values (i.e., recall, precision, and $F_{1.0}$) are averaged over the five runs. Training epochs are 50 for the real data or ten for the pseudo error data at the maximum and we adopt

[4] https://spacy.io/

[5] More generally, it can also be solved as a sequence labeling problem using for example CRF. However, Rei (2017) shows that the grammatical error detection task does not benefit from CRF. We observed the same tendency in our datasets. Accordingly, we solve it as a token classification problem (without CRF).

the epoch achieving the best $F_{1.0}$ on the development set. Other major hyper parameters are shown in Appendix A.

3.2.1 BERT-based Method

The BERT-based method takes as input a word sequence $w_1, \ldots, w_i \ldots, w_N$ and conducts the following procedures:

(1) Subword: convert all w_i into their corresponding subwords: $s_1, \ldots, s_j \ldots, s_M$. Note that the total number of all subwords are generally different from that of all words in the input word sequence.

(2) Encode: encode all s_j into BERT embeddings b_j by:

$$b_j = \text{BERT}(s_j) \quad (1)$$

where BERT$(\cdot)$ denotes BERT taking subwords as input and outputs their corresponding embedding vectors of h-dimension (specifically, $h = 768$ for 'bert-base') from the final layer. We use 'bert-base-uncased' for BERT$(\cdot)$.

(3) Token classification: output the optimal labels by:

$$l_i = \arg\max \text{softmax}(W b_j) \quad (2)$$

where W is a $k \times h$ weight matrix where k is either 2 or $K + 1$ (the number of different labels). To take care of the difference in the lengths of the input word sequence and the corresponding subword sequence, only the first subword of each word is considered in prediction.

3.2.2 Methods to Be Compared

For comparison, we select a BiLSTM-based error detection method. Basically, it follows the above steps (1) to (3), but uses BiLSTM as an encoder in place of BERT. Also, the input word sequence is turned into a sequence of embedding vectors where each embedding vector consists of the concatenation of the conventional word embedding and a character-based embedding. The character-based embedding is obtained by another BiLSTM taking the characters of each word following the work (Akbik et al., 2018). The concatenated embeddings are put into the encoder BiLSTM to produce vectors for prediction in step (3). Specifically, we use the implementation FLAIR (Akbik et al., 2019). We will refer to this method for comparison as the BiLSTM-based method hereafter.

We also investigate how effective the fine-tuning of BERT is. Namely, the BERT part of the BERT-based method is fixed during training and only the output layer is adjusted by the training data. We will refer to this method as the BERT-based method without BERT training hereafter.

4 Performance on Real Data

Figure 1 (see the second page) to Figure 3 show the relationship between the number of training sentences and $F_{1.0}$ in FCE[6], NUCLE, and BEA with its corresponding precision-recall curves, respectively. All $F_{1.0}$ graphs show the high generalization ability of the BERT-based method. They also show that the BERT-based method exhibits a performance saturation at a very early stage (1,000-3,000 training sentences). It is worthwhile to point out the fact that the $F_{1.0}$ curves for all proficiency levels (A, B, and C)[7] in BEA (Figure 3) exhibit the same trend as the other corpora although the absolute performances differ depending on the levels. These results empirically confirm that the high generation ability of the BERT-based method holds in various writer populations.

Unlike the BERT-based method, the BiLSTM-based method improves steadily as the number of training sentences increases although even with the full training data, it only achieves an $F_{1.0}$ that the BERT-based method can achieve with only 500 or less training sentences. Also, the BERT-based method without BERT training does not perform well at all. This is probably because it requires much more degrees of freedom in terms of the network parameters to learn rules for detecting a wide variety of grammatical errors, which have a certain degree of complexity.

To investigate the results from a different point of view, let us now consider precision-recall curves

[6]The best performance of the BERT-based method in FCE is a recall and precision of 0.455 and 0.628, respectively. These numbers are considerably lower than those of the state-of-the-art Yuan et al. (2021) report; they show that the ELECTRA-based method achieves the best recall and precision of 0.505 and 0.821, respectively (c.f., recall: 0.480, precision: 0.757 for their BERT-based method). Note that they use the large models whereas we use the bert-base model, which should be part of the reason for the performance differences. While our results do not achieve the best results, they explain well why large-scale masked language models perform well in grammatical error detection.

[7]The proficiency levels A, B, and C in BEA, which corresponds to those in CEFR, becomes higher in this order.

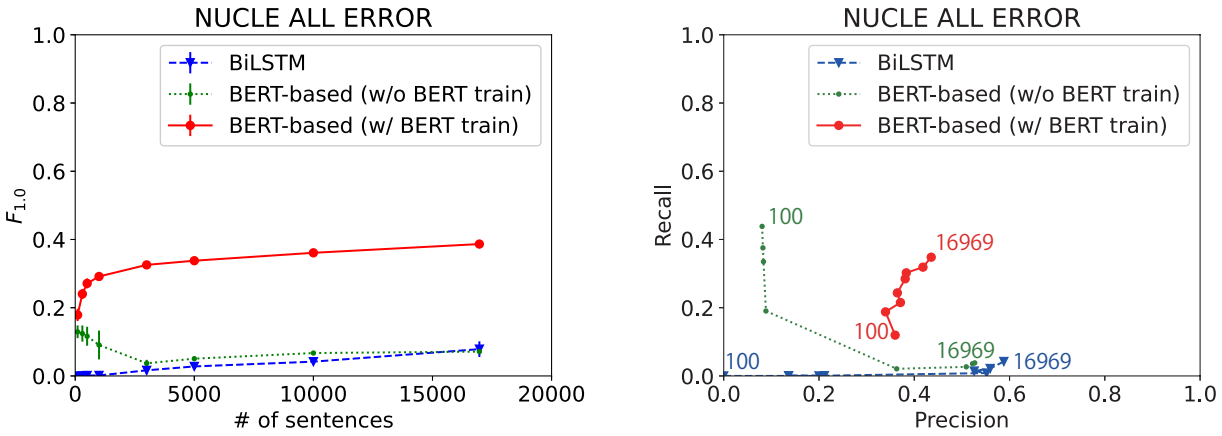

Figure 2: Detection Performance in Relation with Training Size (NUCLE).

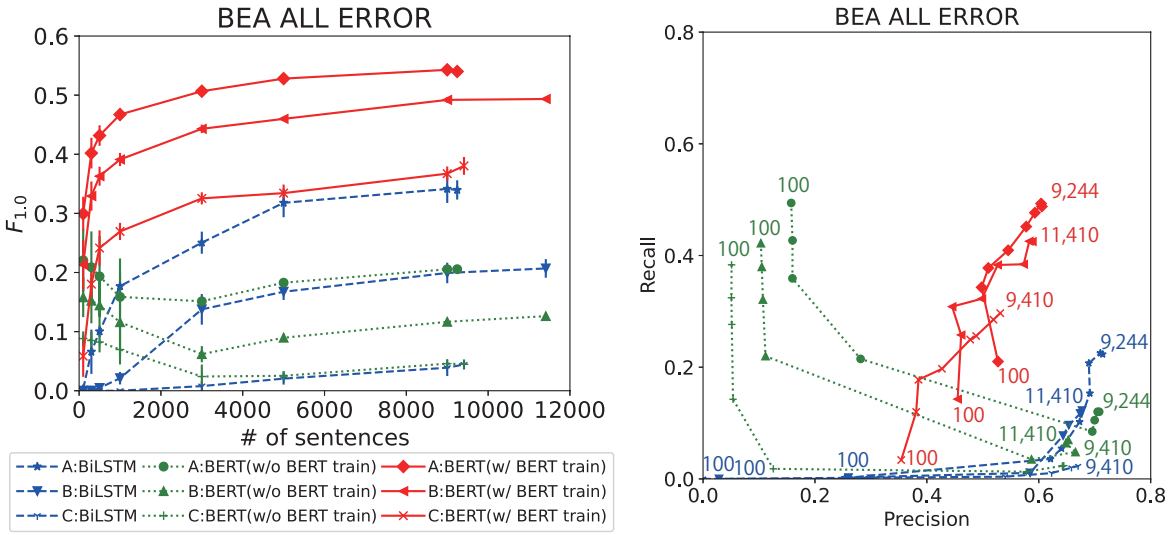

Figure 3: Detection Performance in Relation with Training Size (BEA). Essays in BEA are classified into three categories A, B, and C corresponding to the CEFR levels. The categories are referred to by the labels such as "A:BERT" and "B:BERT" in the legend.

(the right graphs; the numbers at the edges of each curve correspond to the numbers of minimum and maximum training sentences and the plots are lined in minimum to maximum order). All precision-recall curves show that only the BERT-based method quickly improves in recall as the number of training sentences increases while the BERT-based and BiLSTM-based methods both improve in precision. In other words, only the BERT-based method learns to recognize various error types with little exposure to error examples.

Figure 4 shows recalls per detailed error types in FCE[8] where error types are automatically obtained by using ERRANT. Figure 4 shows that the BERT-based method quickly achieves a good recall in SPELL (spelling errors) while the BiLSTM-based method shows a much milder rise. This is not surprising considering that BERT is trained on large native corpora. Namely, it virtually has a large vocabulary list and knows about English spellings well. More interestingly, it exhibits a performance saturation at a very early stage (500-3,000 training sentences) in all errors, resulting in log-like-shape curves while the BiLSTM-based method improves rather linearly (except for SPELL). Even more interestingly, it shows a sharp rise in recall in DET (determiner errors) and NOUN:NUM (errors in noun number). The notion of noun countability with POS plays an important role in detecting these two types of error as in **I am student/countable.* and *an evidence/uncountable*. This suggests that BERT contains some kind of knowledge corresponding to noun countability and POS (singular/plural nouns). More generally, from these, one

[8]PUNCT, OTHER and those whose frequency is less than 150 are excluded in Figure 4.

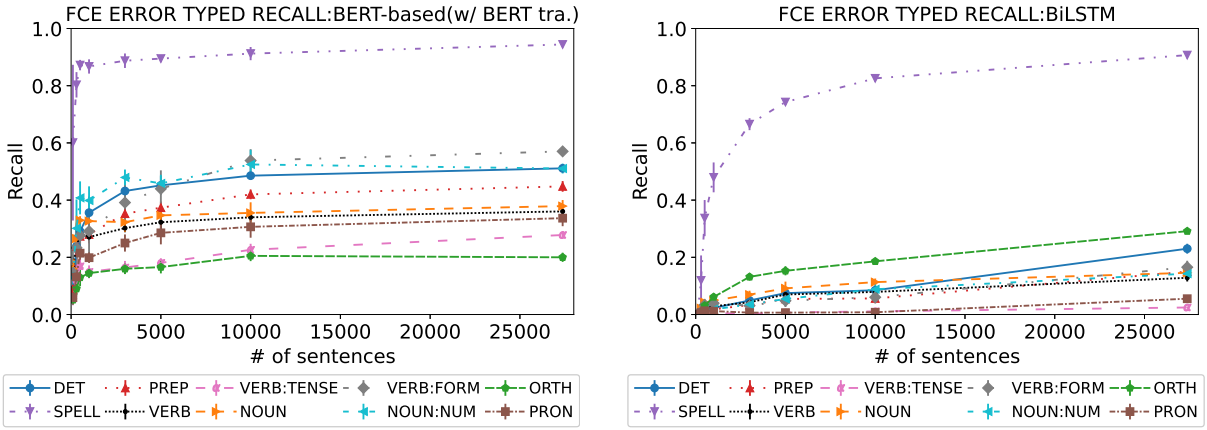

Figure 4: Detection Performance (Recall) per Error Type in Relation with Training Size (FCE).

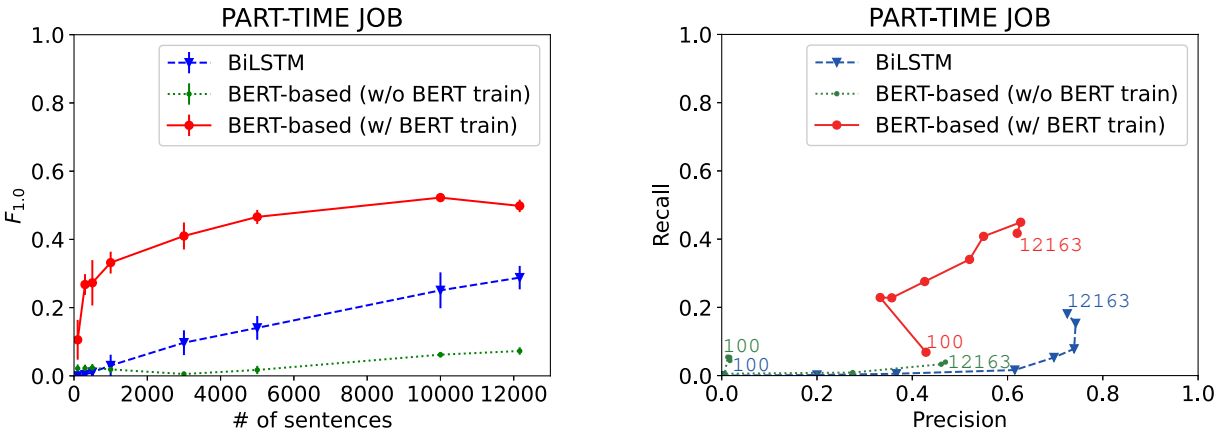

Figure 5: Detection Performance in Relation with Training Size (ICNALE, in-domain test data).

can expect that it has broad knowledge about linguistic properties, which we will observe presently in the experiments with ICNALE.

Let us then turn to ICNALE where we only target errors concerning preposition use (meaning that the other errors are unmarked and that if other errors are detected, they are treated as false positives). Here, we only present performance on PART-TIME JOB due to the space limitation; the results for SMOKING exhibit a very similar tendency, which can be found in Appendix B.

Figure 5 and Figure 6 show performance in PART-TIME JOB in ICNALE; in Figure 5, all models are trained on essays on PART-TIME JOB and tested on (test) essays on the same topic (in-domain setting) while in Figure 6, they are trained on SMOKING and tested on PART-TIME JOB (out-of-domain setting). In both settings, 300 to 500 training sentences are again enough for the BERT-based method to rival the BiLSTM-based method with the full training data, which exhibits again a linear improvement in $F_{1.0}$. Also, only the BERT-based method quickly improves in recall.

A closer look at the detection results reveals that many of the errors require linguistic knowledge including POS, syntactic relations, and lexical properties such as transitive/intransitive verbs. For example, "Preposition + subject" errors, which were introduced in Subsect. 3.1, require the notions of POS such as verbs and syntactic relations such as subjects. Similarly, the distinction between transitive and intransitive verbs play a crucial role in "transitive verb + preposition" and "intransitive verb + object". The fact that the BERT-based method can detect these types of errors with a few training instances suggest that it has an access to grammar-like knowledge and that it can turn the knowledge into error detection rules by fine-tuning. We will explore this in more detail in the following section.

These results also shed light on an important aspect of the BERT-based method in practice. Namely, a cost-effective way of developing an error detection system based on BERT would be to create a small amount of training sentences (e.g., 1,000) for each essay topic; according to the above

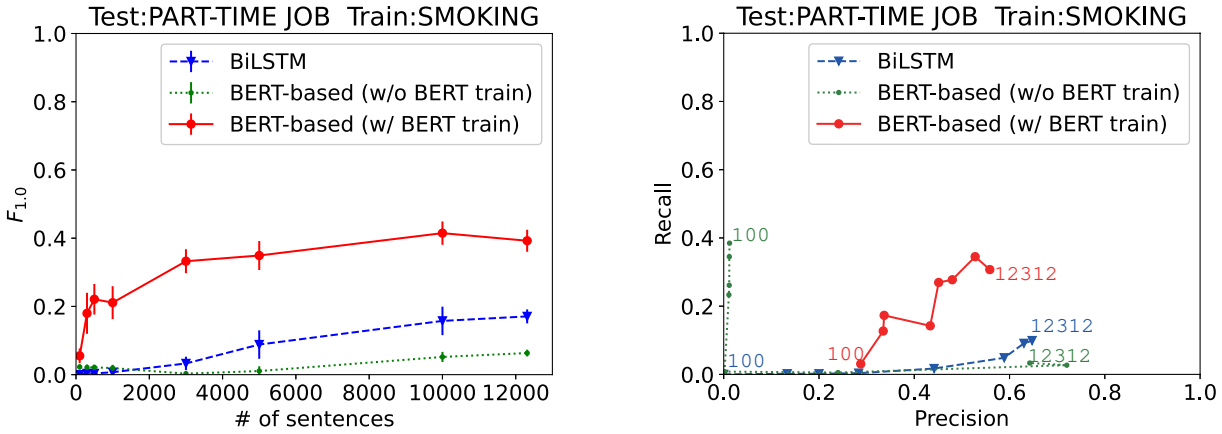

Figure 6: Detection Performance in Relation with Training Size: (ICNALE, out-of-domain test data).

figures, the gain would be much smaller after 1,000 training sentences. Of course, the results are only for a simple BERT-based method. No one knows how differently the performance curves grow with different architectures and/or with a much larger set of training instances. It will be interesting to investigate these points for future work.

5 Performance on Synthetic Data

In the previous section, we have seen that the BERT-based method has a much higher generalization ability in grammatical error detection. To analysis this phenomenon in detail, we now turn to detection performance of the BERT-based method on the pseudo error data. As described in Subsect. 3.1, we train it on the ten sets of training data (i.e., 2, 4, $\cdots$, 1024 training sentences for each error type) and test the trained models on the fixed test set. Doing so, we estimate the relationship between the number of training sentences and detection performance for each error type.

Figure 7 shows the relationship between the size of training data and $F_{1.0}$ for each error type where the size is measured by the number of sentences. Figure 7 reveals that the BERT-based method already recognizes some of the target errors at early stages (even with two or four training sentences). Performance goes much higher even with eight training sentences in most of the error types with an exception of the error type "Intransitive verb + object". For instance, the BERT-based method recognizes more than half of the "Preposition + subject" errors with a precision of 0.800 only with eight training instances. This implies again that BERT has certain knowledge about the notions of POS such as verbs and syntactic relations such as subjects; otherwise, it would be difficult to achieve

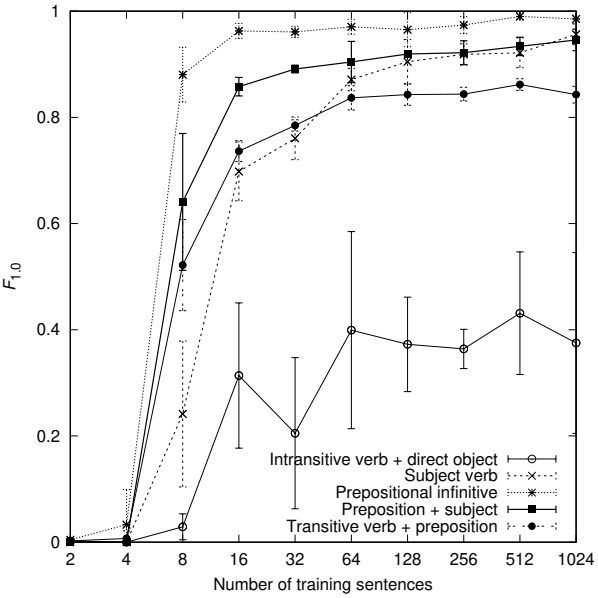

Figure 7: Detection Performance ($F_{1.0}$) per Error Type in Relation with Training Size: Synthetic Data.

a similar performance in this type of error considering that the noun phrase of a subject and its position in the sentence considerably vary depending on the target sentence.

The same argument applies to "transitive verb + preposition" and "intransitive verb + object". It should be emphasized that the BERT-based method has to detect errors in verbs that never[9] appear in the training data; recall that there is no overlap of the target transitive/intransitive verbs in the training and test data. In other words, the BERT-based method can recognize unseen erroneous combinations, for example, **visited in Atlanta* (transitive verb + preposition type) and **specialized environ-*

[9]Strictly, some of the verbs may appear in the training sentences for the other error types. However, they never appear in the erroneous phrases. Also, they do not appear at all when the training size is small.

mental litigation (intransitive verb + direct object error type) after just seeing **mention in*, *discussed about* and **were related drugs* and *belongs Lon's grandmother*. These training and test sentences have almost nothing in common except that they are the combinations of transitive/intransitive verbs and prepositions/objects. Besides, the fact that correct combinations of other verbs and prepositions/objects often appear in the test data makes the task even more difficult without the knowledge of POS and syntactic relations. These findings support the hypotheses that BERT has linguistic knowledge and that it can turn the knowledge into error detection rules by fine-tuning.

6 Exploration for Cost-Effective Error Detection with Feedback Comments

The findings we have obtained so far bring out the possibility that one can implement with few training instances a system that accurately detects grammatical errors and recognizes their detailed error types. For example, manually or automatically, by creating few instances of the erroneous combination of transitive verbs and prepositions as we saw in the previous sections (e.g., **discuss about*), one can develop a system detecting the same type of error in other transitive verbs and prepositions (e.g., **attend in it*). With the detailed error types, the system can also output feedback comments to the user such as *Transitive verbs do not take a preposition. Instead, they take a direct object* instead of just indicating them as preposition errors.

As a pilot study, we trained the BERT-based method on the pseudo error data and tested it on the real (learner) data to examine the above possibility. To achieve it, we manually annotated the real data with the target five error types based on the feedback comments available in ICNALE.

Figure 8 shows the results. It reveals that the BERT-based method trained on the pseudo data does not perform on the real data as well as on the pseudo data. Performance growths stop at an early stage (around eight training sentences). A possible reason for this is that in the real data, multiple errors often appear in a sentence. Also, multiple errors in a sentence can range over multiple types of error. Besides, the error rate is much lower in the real data than in the pseudo error data where one error occurs per sentence except 200 error-free sentences (although multiple types of error appear in the whole data set). These conditions make the task much more difficult on the real data, as Flachs et al. (2019) demonstrate.

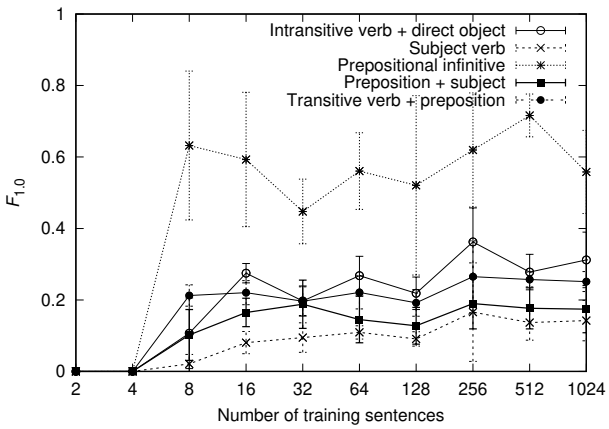

Figure 8: Detection Performance ($F_{1.0}$) on per Error Type in Relation with Training Size: Trained on Synthetic Data; Tested on Real Data (PART-TIME JOB and SMOKING merged).

Having said that, the results shown in Figure 7 still encourage us to develop language model-based systems with a small amount of in-domain training data in order to detect grammatical errors with detailed error types. One possible way to achieve it is to sample sentences from unannotated essays written on the target topic, and then to annotate them with the specific error types that the developer wants to give feedbacks to learners. This will naturally mitigate the problems caused by the multiple-type multiple error situation and the error rate difference. One can also manually create sample error sentences (and their correct versions).

7 Conclusions

In this paper, we have explored the capacity of a large-scale masked language model to recognize grammatical errors. Our findings are summarized in the following three points: (1) Experiments with the real learner data show that a BERT-based error detection method has a much higher generalization ability in grammatical error detection than a non-language model-based method, and the first performance saturation comes at the point of around 1,000-3,000 training instances; (2) It starts to recognize the target errors involving a wide variety of grammatical knowledge with very few instances of them; (3) The high generalization ability brings out its potential for developing systems that detect and explain grammatical errors with very few training instances.

Acknowledgments

This work was partly supported by Japan Science and Technology Agency (JST), PRESTO Grant Number JPMJPR1758, Japan. This work was partly conducted by using computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST).

References

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

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.

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is key: Grammatical error detection with contextual word representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805.

Yong Cheng and Mofan Duan. 2020. Chinese grammatical error detection based on BERT model. In *Proceedings of 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 108–113. Association for Computational Linguistics.

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

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of 8th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bohdan Didenko and Julia Shaptala. 2019. Multi-headed architecture based on BERT for grammatical errors correction. In *Proceedings of 14th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 246–251.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Simon Flachs, Ophélie Lacroix, Marek Rei, Helen Yannakoudakis, and Anders Søgaard. 2019. A simple and robust approach to detecting subject-verb agreement errors. In *Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2418–2427. Association for Computational Linguistics.

David Graff. 2002. The aquaint corpus of english news text.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-head multi-layer attention to deep language representations for grammatical error detection.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254. Association for Computational Linguistics.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? layerwise detection of linguistic anomalies. In *Proceedings of 59th Annual Meeting of the Association for Computational Linguistics and 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4215–4228, Online.

Masato Mita and Hitomi Yanaka. 2021. Do grammatical error correction models realize grammatical generalization? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages

4554–4561. Association for Computational Linguistics.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. Creating corpora for research in feedback comment generation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.

Ryo Nagata and Edward Whittaker. 2013. Reconstructing an Indo-European family tree from non-native English texts. In *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1137–1147, Sofia, Bulgaria. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. In *Proceedings of 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 33–43. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of 33rd Conference on Neural Information Processing Systems*, pages 5753–5763.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Hyper parameter settings

Table 2 shows major hyper parameters used in the experiments. Note that when we use the pseudo error data for training, the number of training sentences can be as small as ten, and we use a rather small batch of five; otherwise we use 32.

Batch size	5 or 32
Optimization	Adam with decoupled weight decay regularization
Learning rate	$5e$-5, (0.9, 0.999)
Epsilon	$1e$-8
Weight decay	$1e$-2

Table 2: Major Hyper parameters.

B Performance on SMOKING in ICNALE

Figure 9 and Figure 10 show performance in SMOKING in ICNALE. We can see the same tendency as in Figures 5 and 6.

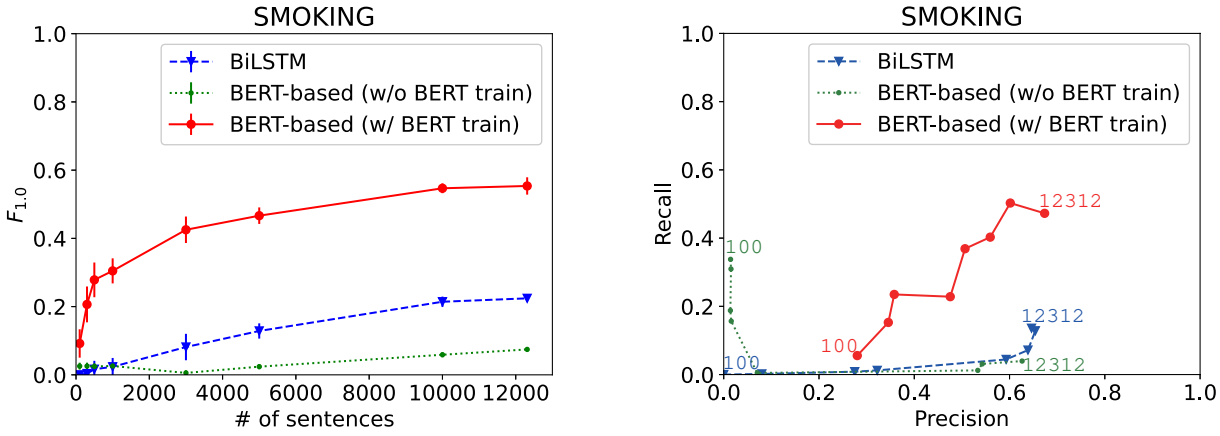

Figure 9: Detection Performance in Relation with Training Size (ICNALE, in-domain test data).

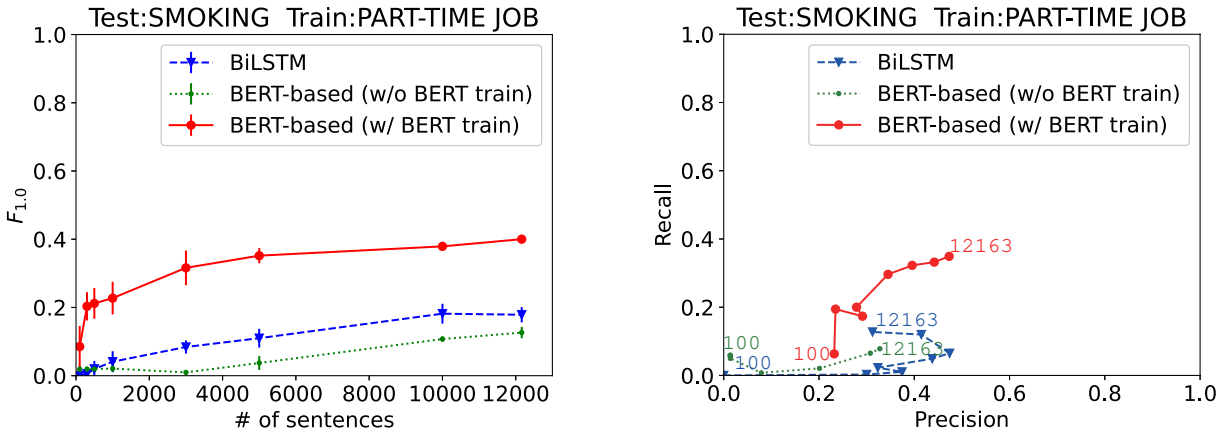

Figure 10: Detection Performance in Relation with Training Size: (ICNALE, out-of-domain test data).