Neural Quality Estimation with Multiple Hypotheses for Grammatical Error Correction

Zhenghao Liu[1,2], Xiaoyuan Yi[1,2], Maosong Sun[1,3]*, Liner Yang[4], Tat-Seng Chua[5]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology
[2]State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
[3]Beijing Academy of Artificial Intelligence
[4]Beijing Language and Culture University, Beijing, China
[5]School of Computing, National University of Singapore, Singapore

Abstract

Grammatical Error Correction (GEC) aims to correct writing errors and help language learners improve their writing skills. However, existing GEC models tend to produce spurious corrections or fail to detect lots of errors. The quality estimation model is necessary to ensure learners get accurate GEC results and avoid misleading from poorly corrected sentences. Well-trained GEC models can generate several high-quality hypotheses through decoding, such as beam search, which provide valuable GEC evidence and can be used to evaluate GEC quality. However, existing models neglect the possible GEC evidence from different hypotheses. This paper presents the Neural Verification Network (VERNet) for GEC quality estimation with multiple hypotheses. VERNet establishes interactions among hypotheses with a reasoning graph and conducts two kinds of attention mechanisms to propagate GEC evidence to verify the quality of generated hypotheses. Our experiments on four GEC datasets show that VERNet achieves state-of-the-art grammatical error detection performance, achieves the best quality estimation results, and significantly improves GEC performance by reranking hypotheses. All data and source codes are available at https://github.com/thunlp/VERNet.

1 Introduction

Grammatical Error Correction (GEC) systems primarily aim to serve second-language learners for proofreading. These systems are expected to detect grammatical errors, provide precise corrections, and guide learners to improve their language ability. With the rapid increase of second-language learners, GEC has drawn growing attention from numerous researchers of the NLP community.
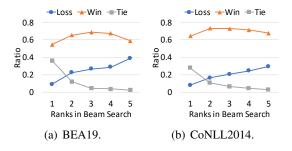

(a) BEA19. (b) CoNLL2014.

Figure 1: The Grammaticality of Generated Hypotheses. The hypotheses are generated by Kiyono et al. (2019) with beam search decoding. The hypothesis is compared to the source sentence with a BERT based language model and classified into Win (the hypothesis is better), Tie (the hypothesis and source are same) and Loss (the source is better). The ratios of different classes are plotted with different beam search ranks.

Existing GEC systems usually inherit the seq2seq architecture (Sutskever et al., 2014) to correct grammatical errors or improve sentence fluency. These systems employ beam search decoding to generate correction hypotheses and rerank hypotheses with quality estimation models from K-best decoding (Kiyono et al., 2019; Kaneko et al., 2020) or model ensemble (Chollampatt and Ng, 2018a) to produce more appropriate and accurate grammatical error corrections. Such models thrive from edit distance and language models (Chollampatt and Ng, 2018a; Chollampatt et al., 2019; Yannakoudakis et al., 2017; Kaneko et al., 2019, 2020). Chollampatt and Ng (2018b) further consider the GEC accuracy in quality estimation by directly predicting the official evaluation metric, $F_{0.5}$ score.

The K-best hypotheses from beam search usually derive from model uncertainty (Ott et al., 2018). These uncertainties of multi-hypotheses come from model confidence and potential ambiguity of lin-

---

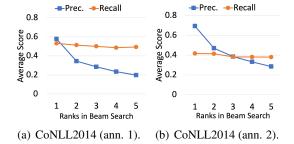(a) CoNLL2014 (ann. 1).   (b) CoNLL2014 (ann. 2).

Figure 2: The GEC Performance of Generated Hypotheses. The hypotheses generated by Kiyono et al. (2019) are evaluated on the CoNLL2014 dataset. The average scores of Precision and Recall are calculated according to the two annotations of CoNLL2014.

guistic variation (Fomicheva et al., 2020), which can be used to improve machine translation performance (Wang et al., 2019b). Fomicheva et al. (2020) further leverage multi-hypotheses to make convinced machine translation evaluation, which is more correlated with human judgments. Their work further demonstrates that multi-hypotheses from well-trained neural models have the ability to provide more hints to estimate generation quality.

For GEC, the hypotheses from the beam search decoding of well-trained GEC models can provide some valuable GEC evidence. We illustrate the reasons as follows.

- *Beam search can provide better GEC results.* The GEC performance of the top-ranked hypothesis and the best one has a large gap in beam search. For two existing GEC systems, Zhao et al. (2019) and Kiyono et al. (2019), the $F_{0.5}$ scores of these systems are 58.99 and 62.03 on the CoNLL2014 dataset. However, the $F_{0.5}$ scores of the best GEC results of these systems can achieve 73.56 and 76.82.

- *Beam search candidates are more grammatical.* As shown in Figure 1, the hypotheses from well-trained GEC models with beam search usually win the favor of language models, even for these hypotheses ranked to the rear. It illustrates these hypotheses are usually more grammatical than source sentences.

- *Beam search candidates can provide valuable GEC evidence.* As shown in Figure 2, the hypotheses of different beam ranks have almost the same Recall score, which demonstrates all hypotheses in beam search can provide some valuable GEC evidence.

Existing quality estimation models (Chollampatt and Ng, 2018b) for GEC regard hypotheses independently and neglect the potential GEC evidence from different hypotheses. To fully use the valuable GEC evidence from GEC hypotheses, we propose the Neural Verification Network (VERNet) to estimate the GEC quality with modeled interactions from multi-hypotheses. Given a source sentence and K hypothesis sentences from the beam search decoding of the basic GEC model, VERNet establishes hypothesis interactions by regarding ⟨source, hypothesis⟩ pairs as nodes, and constructing a fully-connected reasoning graph to propagate GEC evidence among multi-hypotheses. Then VERNet proposes two kinds of attention mechanisms on the reasoning graph, *node interaction attention* and *node selection attention*, to summarize and aggregate necessary GEC evidence from other hypotheses to estimate the quality of tokens.

Our experiments show that VERNet can pick up necessary GEC evidence from multi-hypotheses provided by GEC models and help verify the quality of GEC hypotheses. VERNet helps GEC models to generate more accurate GEC results and benefits most grammatical error types.

2 Related Work

The GEC task is designed for automatically proof-reading. Large-scale annotated corpora (Mizumoto et al., 2011; Dahlmeier et al., 2013; Bryant et al., 2019) bring an opportunity for building fully data-driven GEC systems.

Existing neural models regard GEC as a natural language generation (NLG) task and usually use sequence-to-sequence architecture (Sutskever et al., 2014) to generate correction hypotheses with beam search decoding (Yuan and Briscoe, 2016; Chollampatt and Ng, 2018a). Transformer-based architectures (Vaswani et al., 2017) show their effectiveness in NLG tasks and are also employed to achieve convinced correction results (Grundkiewicz et al., 2019; Kiyono et al., 2019). The copying mechanism is also introduced for GEC models (Zhao et al., 2019) to better align tokens from source sentence to hypothesis sentence. To further accelerate the generation process, some work also comes up with non-autoregressive GEC models and leverages a single encoder to parallelly detect and correct grammatical errors (Awasthi et al., 2019; Malmi et al., 2019; Omelianchuk et al., 2020).

Recent research focuses on two directions to im-

prove GEC systems. The first one treats GEC as a low-resource language generation problem and focuses on data augmentation for a grammar sensitive and language proficient GEC system (Junczys-Dowmunt et al., 2018; Kiyono et al., 2019). Various weak-supervision corpora have been leveraged, such as Wikipedia edit history (Lichtarge et al., 2019), Github edit history (Hagiwara and Mita, 2020) and confusing word set (Grundkiewicz et al., 2019). Besides, lots of work generates grammatical errors through generation models or round-trip translation (Ge et al., 2018; Wang et al., 2019a; Xie et al., 2018). Kiyono et al. (2019) further consider different data augmentation strategies to conduct better GEC pretraining.

Reranking GEC hypotheses from K-best decoding or GEC model ensemble (Hoang et al., 2016; Chollampatt and Ng, 2018b) with quality estimation models provides another promising direction to achieve better GEC performance. Some methods evaluate if hypotheses satisfy linguistic and grammatical rules. For this purpose, they employ language models (Chollampatt and Ng, 2018a; Chollampatt et al., 2019) or grammatical error detection (GED) models to estimate hypothesis quality. GED models (Rei, 2017; Rei and Søgaard, 2019) estimate the hypothesis quality on both sentence level (Kaneko et al., 2019) and token level (Yannakoudakis et al., 2017). Chollampatt and Ng (2018b) further estimate GEC quality by considering correction accuracy. They establish source-hypothesis interactions with the encoder-decoder architecture and learn to directly predict the official evaluation score $F_{0.5}$.

The pre-trained language model BERT (Devlin et al., 2019) has proven its effectiveness in producing contextual token representations, achieving better quality estimation (Kaneko et al., 2019; Chollampatt et al., 2019) and improving GEC performance by fuse BERT representations (Kaneko et al., 2020). However, existing quality estimation models regard each hypothesis independently and neglect the interactions among multi-hypotheses, which can also benefit the quality estimation (Fomicheva et al., 2020).

3 Neural Verification Network

This section describes Neural Verification Network (VERNet) to estimate the GEC quality with multi-hypotheses, as shown in Figure 3.

Given a source sentence s and K correspond-

Source Sentence:
s: Do one who suffered from this disease ...

Hypotheses from the beam search decoding of basic GEC :
c^1: Does someone who suffered from this disease ...
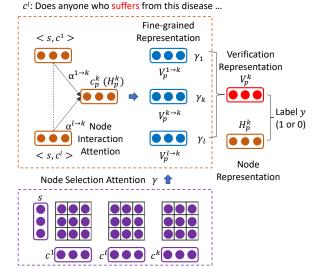c^k: Does someone who suffers (p-th token) from this disease ...
c^l: Does anyone who suffers from this disease ...

Figure 3: The Architecture of Neural Verification Network (VERNet). The underlined estimated token (c_p^k) and potentially **supporting evidence** towards c_p^k are annotated.

ing hypotheses $C = \{c^1, \ldots, c^k, \ldots, c^K\}$ generated by a GEC model, we first regard each source-hypothesis pair $\langle s, c^k \rangle$ as a node and fully connect all nodes to establish multi-hypothesis interactions. Then VERNet leverages BERT to get the representation of each token in $\langle s, c^k \rangle$ pairs (Sec. 3.1) and conducts two kinds of attention mechanisms to propagate and aggregate GEC evidence from other hypotheses to verify the token quality (Sec. 3.2). Finally, VERNet estimates hypothesis quality by aggregating token level quality estimation scores (Sec. 3.3). Our VERNet is trained end-to-end with supervisions from golden labels (Sec. 3.4).

3.1 Initial Representations for Sentence Pairs

Pre-trained language models, *e.g.* BERT (Devlin et al., 2019), show their advantages of producing contextual token representations for various NLP tasks. Hence, given a source sentence s with m tokens and the k-th hypothesis c^k with n tokens, we use BERT to encode the source-hypothesis pair $\langle s, c^k \rangle$ and get its representation H^k:

$$H^k = \text{BERT}([\text{CLS}]\ s\ [\text{SEP}]\ c^k\ [\text{SEP}]). \quad (1)$$

The pair representation H^k consists of token-level representations, that is,

$H^k = \{H_0^k, \ldots, H_{m+n+2}^k\}$. H_0^k denotes the representation of "[CLS]" token.

3.2 Verify Token Quality with Multi-hypotheses

VERNet conducts two kinds of attention mechanisms, *node interaction attention* and *node selection attention*, to verify the token quality with the verification representation V^k of k-th node, which learns the supporting evidence towards estimating token quality from multi-hypotheses.

The node interaction attention first summarizes useful GEC evidence from the l-th node for the fine-grained representation $V^{l \to k}$ (Sec. 3.2.1). Then node selection attention further aggregates fine-grained representation $V^{l \to k}$ with score γ^l according to each node's confidence (Sec. 3.2.2). Finally, we can calculate the verification representation V^k to verify the token's quality of each node.

3.2.1 Fine-grained Node Representation with Node Interaction Attention

The node interaction attention $\alpha^{l \to k}$ attentively reads tokens in the l-th node and picks up supporting evidence towards the k-th node to build fine-grained node representations $V^{l \to k}$.

For the p-th token in the k-th node, w_p^k, we first calculate the node interaction attention weight $\alpha_q^{l \to k}$ according to the relevance between w_p^k and the q-th token in the l-th node, w_q^l:

$$\alpha_q^{l \to k} = \text{softmax}_q((H_p^k)^T \cdot W \cdot H_q^l), \qquad (2)$$

where W is a parameter. H_p^k and H_q^l are the representations of w_p^k and w_q^l. Then all token representations of l-th node are aggregated:

$$V_p^{l \to k} = \sum_{q=1}^{m+n+2} (\alpha_q^{l \to k} \cdot H_q^l). \qquad (3)$$

Based on $V_p^{l \to k}$, we further build the l-th node fine-grained representation towards the k-th node, $V^{l \to k} = \{V_1^{l \to k}, \ldots, V_p^{l \to k}, \ldots, V_{m+n+2}^{l \to k}\}$.

3.2.2 Evidence Aggregation with Node Selection Attention

The node selection attention measures node importance and is used to aggregate supporting evidence from the fine-grained node representation $V^{l \to k}$ of the l-th node. We leverage attention-over-attention mechanism (Cui et al., 2017) to conduct source h^{ls} and hypotheses h^{lh} representations to calculate the l-th node selection attention score γ^l. Then we get

the node verification representation V_p^k with the node selection attention γ^l.

To calculate the node selection attention γ^l, we establish an interaction matrix M^l between the source and hypothesis sentences of the l-th node. Each element M_{ij}^l in M^l is calculated with the relevance between i-th source token and j-th hypothesis token (include "[SEP]" tokens):

$$M_{ij}^l = (H_i^l)^T \cdot W \cdot H_{m+1+j}^l, \qquad (4)$$

where W is a parameter. Then we calculate attention scores β_i^{ls} and β_j^{lh} along the source dimension and hypothesis dimension, respectively:

$$\beta_i^{ls} = \frac{1}{n+1} \sum_{j=1}^{n+1} \text{softmax}_i(M_{ij}^l), \qquad (5)$$

$$\beta_j^{lh} = \frac{1}{m+1} \sum_{i=1}^{m+1} \text{softmax}_j(M_{ij}^l). \qquad (6)$$

Then the representations of source sentence and hypothesis sentence are calculated:

$$h^{ls} = \sum_{i=1}^{m+1} \beta_i^{ls} \cdot H_i^l, \quad h^{lh} = \sum_{j=1}^{n+1} \beta_j^{lh} \cdot H_{m+1+j}^l. \qquad (7)$$

Finally, the node selection attention γ^l of l-th node is calculated for the evidence aggregation:

$$\gamma^l = \text{softmax}_l(\text{Linear}((h^{ls} \circ h^{lh}); h^{ls}; h^{lh})), \qquad (8)$$

where $\circ$ is the element-wise multiplication operator and ; is the concatenate operator.

The node selection attention γ^l aggregates evidence for the verification representation V_p^k of w_p^k:

$$V_p^k = \sum_{l=1}^{K} (\gamma^l \cdot V_p^{l \to k}), \qquad (9)$$

where $V^k = \{V_1^k, \ldots, V_p^k, \ldots, V_{m+n+2}^k\}$ is the k-th node verification representation.

3.3 Hypothesis Quality Estimation

For the p-th token w_p^k in the k-th node, the probability $P(y|w_p^k)$ of quality label y is calculated with the verification representation V_p^k:

$$P(y|w_p^k) = \text{softmax}_y(\text{Linear}((H_p^k \circ V_p^k); H_p^k; V_p^k)), \quad (10)$$

where $\circ$ is the element-wise multiplication and ; is the concatenate operator. We average all probability $P(y=1|w_p^k)$ of token level quality estimation as hypothesis quality estimation score $f(s, c^k)$ for the pair $\langle s, c^k \rangle$:

$$f(s, c^k) = \frac{1}{n+1} \sum_{p=m+2}^{m+n+2} P(y=1|w_p^k). \qquad (11)$$

3.4 End-to-end Training

We conduct joint training with token-level supervision. The source labels and hypothesis labels are used, which denote the grammatical quality of source sentences and GEC accuracy of hypotheses.

The cross entropy loss for the p-th token w_p^k in the k-th node is calculated:

$$L(w_p^k) = \text{CrossEntropy}(y^*, P(y|w_p^k)), \qquad (12)$$

using the ground truth token labels y^*.

Then the training loss of VERNet is calculated:

$$L = \frac{1}{K} \frac{1}{m+n+2} \sum_{k=1}^{K} \sum_{p=1}^{m+n+2} L(w_p^k). \qquad (13)$$

4 Experimental Methodology

This section describes the datasets, evaluation metrics, baselines, and implementation details.

Datasets. We use FCE (Yannakoudakis et al., 2011), BEA19 (Bryant et al., 2019) and NU-CLE (Dahlmeier et al., 2013) to construct training and development sets. Four testing scenarios, FCE, BEA19 (Restrict), CoNLL-2014 (Ng et al., 2014) and JFLEG (Napoles et al., 2017), are leveraged to evaluate model performance. Detailed data statistics are presented in Table 1. We do not incorporate additional training corpora for fair comparison.

Basic GEC Model. To generate correction hypotheses, we take one of the state-of-the-art autoregressive GEC systems (Kiyono et al., 2019) as our *basic GEC model* and keep the same setting. The beam size of our baseline model is set to 5 (Kiyono et al., 2019), and all these beam search hypotheses are reserved in our experiments.

We generate quality estimation labels for tokens in both source sentences and hypothesis sentences with ERRANT (Bryant et al., 2017; Felice et al., 2016), which indicate grammatical correctness and GEC accuracy, respectively. As shown in Table 2, ERRANT annotates edit operations (delete, insert, and replace) towards the ground truth corrections. In terms of such annotations, each token is labeled with correct (1) or incorrect (0).

Evaluation Metrics. We introduce the evaluation metrics in three tasks: token quality estimation, sentence quality estimation, and GEC.

To evaluate the model performance of token-level quality estimation, we employ the same evaluation metrics from previous GED models (Rei, 2017; Rei and Søgaard, 2019; Yannakoudakis et al., 2017), including Precision, Recall, and $F_{0.5}$. $F_{0.5}$ is our primary evaluation metric.

Dataset	Training	Development	Test
FCE	28,350	2,191	2,695
BEA19	34,308	4,384	4,477
NUCLE	57,151	-	-
CoNLL-2014	-	-	1,312
JFLEG	-	-	747
Total	119,809	6,575	9,231

Table 1: Data Statistics.

Sentence	The $_1$ a $_2$ Mobile phone is a marvelous invention to $_9$ charge $_{10}$ the world $_{12}$ [SEP]		
	Operation	Span	Edit
Correction	Delete	1,2	-
	Replace	9,10	change
	Insert	12,12	.

Table 2: An Example of Token Label Annotation. All sentences are annotated with ERRANT according to the golden correction. The words in red color are labeled as incorrect (0) and others are labeled as correct (1). The "[SEP]" token denotes the end of the sentence.

For the evaluation of sentence-level quality estimation, we employ the same evaluation metrics from the previous quality estimation model (Chollampatt and Ng, 2018b), including two evaluation scenarios: (1) GEC evaluation metrics for the hypothesis that reranked top-1 and (2) Pearson Correlation Coefficient (PCC) between reranking scores and golden scores ($F_{0.5}$) for all hypotheses.

To evaluate GEC performance, we adopt GLEU (Napoles et al., 2015) to evaluate model performance on the JFLEG dataset. The official tool ERRANT of the BEA19 shared task (Bryant et al., 2019) is used to calculate Precision, Recall, and $F_{0.5}$ scores for other datasets. For the CoNLL-2014 dataset, the M^2 evaluation (Dahlmeier and Ng, 2012) is also adopted as our main evaluation.

Baselines. BERT-fuse (GED) (Kaneko et al., 2020) is compared in our experiments, which trains BERT with the GED task and fuses BERT representations into the Transformer. For quality estimation, we consider two groups of baseline models in our experiments, and more details of these models can be found in Appendices A.1.

(1) *BERT based language models.* We employ three BERT based language models to estimate the quality of hypotheses. BERT-LM (Chollampatt et al., 2019) measures hypothesis quality with the perplexity of the language model. BERT-GQE (Kaneko et al., 2019) is trained with annotated GEC data and estimates if the hypothesis has grammatical errors. We also conduct BERT-GED (SRC) that predicts token level grammar indicator

labels, which is inspired by GED models (Yannakoudakis et al., 2017). BERT shows significant improvement compared to LSTM based models for the GED task (Appendices A.2). Hence the LSTM based models are neglected in our experiments.

(2) *GEC accuracy estimation models.* These models further consider the source-hypothesis interactions to evaluate GEC accuracy. We take a strong baseline NQE (Chollampatt and Ng, 2018b) in experiments. NQE employs the encoder-decoder (predictor) architecture to encode source-hypothesis pairs and predicts $F_{0.5}$ score with the estimator architecture. All their proposed architectures, NQE (CC), NQE (RC), NQE (CR), and NQE (RR) are compared. For NQE (XY), X indicates the predictor architecture, and Y indicates the estimator architecture. X and Y can be recurrent (R) or convolutional (C) neural networks. In addition, we also employ BERT to encode source-hypothesis pairs and then predict the $F_{0.5}$ score to implement the BERT-QE model. We also come up with two baselines, BERT-GED (HYP) and BERT-GED (JOINT). They leverage BERT to encode source-hypothesis pairs and are supervised with the token-level quality estimation label. BERT-GED (HYP) is trained with the supervision of hypotheses, and BERT-GED (JOINT) is supervised with labels from both source and hypothesis sentences.

Implementation Details. In all experiments, we use the base version of BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020). BERT is a widely used pretrained language model and trained with the mask language model task. ELECTRA is trained with the replaced token detection task and aims to predict if the token is original or replaced by a BERT based generator during pre-training. ELECTRA is a discriminator based pretrained language model and is more like the GED task. We regard BERT as our main model for text encoding and leverage ELECTRA to evaluate the generalization ability of our model.

Both BERT and ELECTRA inherit huggingface's PyTorch implementation (Wolf et al., 2020). Adam (Kingma and Ba, 2015) is utilized for parameter optimization. We set the max sentence length to 120 for source and hypothesis sentences, learning rate to 5e-5, batch size to 8, and accumulate step to 4 during training.

For hypothesis reranking, we leverage the learning-to-rank method, Coordinate Ascent (CA) (Metzler and Croft, 2007), to aggregate the ranking features and basic GEC score to conduct the ranking score. We assign the hypotheses with the highest $F_{0.5}$ score as positive instances and the others as negative ones. The Coordinate Ascent method is implemented by RankLib[1].

5 Evaluation Results

We conduct experiments to study the performance of VERNet from three aspects: token-level quality estimation, sentence-level quality estimation, and the VERNet's effectiveness in GEC models. Then we present the case study to qualitatively analyze the effectiveness of the proposed two types of attention in VERNet.

5.1 Performance of Token Level Quality Estimation

We first evaluate VERNet's effectiveness on token-level quality estimation. BERT-GED (SRC) is the previous state-of-the-art GED model (Kaneko and Komachi, 2019). Additional two variants, HYP and JOINT, of BERT-GED are conducted as baselines by considering the first-ranked GEC hypothesis in beam search decoding.

As shown in Table 3, there are two scenarios, *source* and *hypothesis*, are conducted to evaluate model performance. The *source scenario* evaluates the ability of grammaticality quality estimation, which is the same as GED models (Rei and Søgaard, 2019). The *hypothesis scenario* tests the quality estimation ability on GEC accuracy.

For the *source scenario*, BERT-GED (JOINT) outperforms BERT-GED (SRC) and illustrates that the GEC result can help estimate the grammaticality quality of source sentences. For the *hypothesis scenario*, BERT-GED (JOINT) shows better performance than BERT-GED (HYP), which thrives from the supervisions from source sentences. For *both scenarios*, BERT-VERNet shows further improvement compared with BERT-GED (JOINT). Such improvements demonstrate that various GEC evidence from multiple hypotheses benefits the token-level quality estimation.

Moreover, the detection style pre-trained model ELECTRA (Clark et al., 2020) is also used as our sentence encoder. VERNet is boosted a lot on all scenarios and datasets, which illustrates the strong ability of ELECTRA in token-level quality estimation and the generalization ability of VERNet.

---

[1]https://sourceforge.net/p/lemur/wiki/RankLib/

	Model	FCE test set			CoNLL-2014 ann. 1			CoNLL-2014 ann. 2		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Source	BERT-GED (SRC)	74.22	43.34	64.97	59.84	27.11	48.20	77.94	25.02	54.77
	BERT-GED (JOINT)	75.62	44.44	66.32	60.79	27.33	48.83	77.42	25.23	54.77
	BERT-VERNet	**81.53**	45.71	70.48	**62.64**	30.62	51.80	**82.25**	28.49	59.71
	ELECTRA-VERNet	80.94	**50.51**	**72.24**	62.50	**35.61**	**54.30**	81.69	**32.97**	**63.06**
Hypothesis	BERT-GED (HYP)	80.27	40.58	67.14	74.28	34.20	60.17	66.49	27.68	51.93
	BERT-GED (JOINT)	76.71	46.94	68.07	71.15	38.30	60.73	64.79	31.52	53.50
	BERT-VERNet	**81.85**	44.27	69.97	**76.03**	34.02	60.97	71.79	29.04	55.46
	ELECTRA-VERNet	80.62	**49.16**	**71.48**	74.80	**39.26**	**63.33**	72.55	**34.42**	**59.39**

Table 3: Performance of Token Level Quality Estimation. Both source and hypothesis scenarios are conducted to evaluate grammatical quality estimation ability on source sentences and GEC quality estimation ability on hypotheses, respectively. BERT-GED (SRC) only encodes source sentences while others encode ⟨source, hypothesis⟩ pairs. BERT-GED (JOINT) is supervised with golden labels from source and hypothesis sentences.

Model	CoNLL-2014 (M^2)					FCE				BEA19			JFLEG	
	P	R	$F_{0.5}$	PCC (ann.1)	PCC (ann.2)	P	R	$F_{0.5}$	PCC	P	R	$F_{0.5}$	GLEU	PCC
NQE (RR)	61.38	33.03	52.39	23.43	6.62	51.43	30.36	45.16	28.74	57.22	46.33	54.65	55.90	1.29
NQE (RC)	60.09	33.11	51.67	24.12	5.52	53.97	31.35	47.17	31.20	57.87	47.24	55.37	56.91	1.66
NQE (CR)	62.52	35.24	54.14	24.80	9.12	51.77	31.46	45.85	30.69	57.92	47.43	55.47	56.92	6.48
NQE (CC)	60.62	35.77	53.23	22.94	8.39	50.21	32.09	45.11	29.23	56.83	49.47	55.19	57.22	7.68
BERT-LM	52.82	49.59	52.14	3.47	17.62	36.97	43.42	38.10	8.59	46.32	64.05	49.03	59.72	26.85
BERT-GQE	52.67	50.39	52.19	2.56	14.54	36.05	43.53	37.33	10.18	46.15	64.01	48.88	60.17	29.05
BERT-GED (SRC)	52.98	**52.07**	52.79	3.78	20.56	37.58	**45.81**	38.98	12.71	47.15	**65.09**	49.90	60.32	27.28
BERT-QE	62.24	38.27	55.31	22.85	12.17	52.01	36.89	48.07	33.84	58.63	54.19	57.69	59.73	26.16
BERT-GED (HYP)	68.90	34.35	57.36	30.06	16.79	57.21	36.03	51.19	43.48	68.18	53.85	64.73	60.00	29.90
BERT-GED (JOINT)	69.33	36.02	58.51	28.62	16.28	58.53	37.24	52.53	45.08	66.80	55.09	64.07	60.49	33.03
BERT-VERNet	68.75	40.26	60.22	31.02	22.75	58.32	39.99	53.42	47.19	66.86	58.60	65.02	61.36	36.98
ELECTRA-VERNet	**69.97**	42.12	**61.80**	**37.18**	**28.77**	**58.77**	41.86	**54.37**	**48.12**	**69.09**	60.91	**67.28**	**61.61**	**38.63**

Table 4: Performance of Sentence Level Quality Estimation. The ranked top-1 hypothesis is used to calculate GEC metrics. NQE (Chollampatt and Ng, 2018b) uses RNN or CNN models for GEC quality estimation. BERT-LM (Chollampatt et al., 2019) measures perplexity without fine-tuning. BERT-GQE (Kaneko et al., 2019) and BERT-GED (SRC) are supervised with sentence-level and token-level labels from source sentences to estimate grammatical quality, respectively. NQE and BERT-QE encode ⟨source, hypothesis⟩ pairs and directly predict $F_{0.5}$ score. BERT-GED (HYP) and BERT-GED (JOINT) encode the ⟨source, hypothesis⟩ pairs to estimate the quality of generated tokens.

5.2 Performance of Sentence Level Quality Estimation

In this part, we evaluate VERNet's performance on sentence-level quality estimation by reranking hypotheses from beam search decoding.

Baselines can be divided into two groups: language model based and GEC accuracy based quality estimation models. The former focuses on grammaticality and fluency, including BERT-LM, BERT-GQE and BERT-GED (SRC). The others focus on estimating the GEC accuracy, including NQE, BERT-QE, BERT-GED (HYP)/(JOINT).

As shown in Table 4, we find that language model based quality estimation prefers higher recall but lower precision, which leads to more redundant corrections. Only considering grammaticality is insufficient since such unnecessary correction suggestions may mislead users. By contrast, GEC accuracy based quality estimation models get much better Precision and $F_{0.5}$, and provide more precise feedback for users. Furthermore, BERT-GED (HYP) outperforms BERT-QE, manifesting that token-level supervisions provide finer-granularity signals to help the model better distinguish subtle differences among hypotheses. VERNet outperforms all baselines, which supports our claim that multi-hypotheses from beam search provide valuable GEC evidence and help conduct more effective quality estimation for generated GEC hypotheses.

5.3 VERNet's Effectiveness in GEC Models

This part explores the effectiveness of VERNet on improving GEC models. We conduct VERNet[†] by aggregating scores from the basic GEC model and VERNet for hypothesis reranking.

As shown in Table 5, two baseline models are compared in our experiments, Basic GEC (Kiyono et al., 2019) and BERT-fuse (GED) (Kaneko et al., 2020). Compared to BERT-fuse (GED), BERT-VERNet[†] achieves comparable performance on CoNLL-2014 and more improvement on BEA19. It demonstrates that reranking hypotheses with VER-

Model	CoNLL-2014 (M^2)			CoNLL-2014			FCE			BEA19			JFLEG
	P	R	F$_{0.5}$	P	R	F$_{0.5}$	P	R	F$_{0.5}$	P	R	F$_{0.5}$	GLEU
Basic GEC	68.59	44.87	62.03	64.26	43.59	58.69	55.11	41.61	51.75	66.20	**61.48**	65.20	61.00
Basic GEC w. R2L*	72.4	46.1	65.0	-	-	-	-	-	-	**74.7**	56.7	**70.2**	61.4
BERT-fuse (GED)	69.2	45.6	62.6	-	-	-	-	-	-	67.1	60.1	65.6	61.3
BERT-fuse (GED) w. R2L*	**72.6**	**46.4**	**65.2**	-	-	-	-	-	-	72.3	61.4	69.8	**62.0**
BERT-VERNet[†] (Top2)	69.98	**43.69**	62.47	65.62	**41.98**	58.98	58.57	41.53	54.13	68.42	**60.32**	66.63	61.17
BERT-VERNet[†] (Top3)	70.49	43.16	**62.57**	65.92	41.22	58.86	59.20	41.53	54.55	69.03	60.20	67.06	**61.24**
BERT-VERNet[†] (Top4)	**70.79**	42.72	62.56	**66.65**	40.94	**59.21**	59.55	**41.55**	54.80	**69.43**	60.17	**67.36**	61.16
BERT-VERNet[†] (Top5)	70.60	42.50	62.36	66.41	40.74	58.98	**59.68**	41.48	**54.86**	69.39	60.12	67.32	61.10
ELECTRA-VERNet[†] (Top2)	71.21	**44.24**	63.47	66.95	**42.97**	60.22	58.31	41.97	54.09	69.27	61.22	67.50	61.60
ELECTRA-VERNet[†] (Top3)	**71.87**	44.13	**63.84**	67.51	42.38	**60.35**	59.02	41.99	54.59	70.64	61.78	68.67	61.80
ELECTRA-VERNet[†] (Top4)	71.85	43.81	63.69	67.48	42.19	60.25	59.65	42.12	55.07	**70.96**	**62.03**	**68.98**	**62.05**
ELECTRA-VERNet[†] (Top5)	71.58	43.57	63.43	67.15	42.10	60.01	**59.95**	**42.19**	**55.29**	70.79	61.74	68.77	62.07

Table 5: Performance of Hypothesis Reranking. BERT/ELECTRA-VERNet[†] aggregates the scores of Basic GEC Model (Kiyono et al., 2019) and VERNet for hypothesis reranking with Coordinate Ascent. BERT-fuse (GED) (Kaneko et al., 2020) is the Transformer model that fuses BERT representations. *Note that R2L models incorporate four right-to-left Transformer models that are trained with unpublished data and these models are not supplied in their open source codes, thus these results are hard to reimplement.
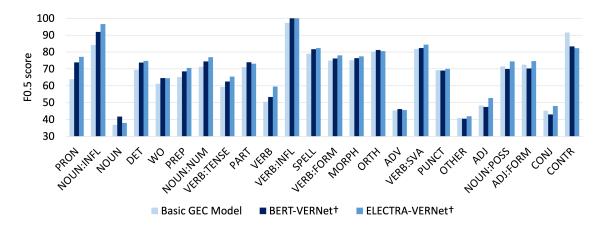

Figure 4: Model Performance of Different Grammatical Error Types on BEA19. VERNet[†] reranks hypotheses with the aggregated score of basic GEC model and VERNet. All types are from ERRANT (Bryant et al., 2017).

Net provides an effective way to improve basic GEC model performance without changing the Transformer architecture. R2L models incorporate four right-to-left Transformer models to improve GEC performance. However, these R2L models are not available. ELECTRA-VERNet[†] incorporates only one model and achieves comparable performance on BEA19 and JFLEG.

Figure 4 presents VERNet[†]'s performance on different grammatical error types. We plot the F$_{0.5}$ scores of both basic GEC model and VERNet[†] on BEA19. VERNet[†] achieves improvement on most types and performs significantly better for word morphology and word usage errors, such as Noun Inflection (NOUN:INFL) and Pronoun (PRON). Such results illustrate that VERNet[†] is able to leverage clues learned from multi-hypotheses to verify the GEC quality. However, we also find that VERNet[†] discounts GEC performance on a few

error types, e.g., Contraction (CONTR). The annotation biases may cause such a decrease in CONTR errors. For example, for both "n't" and "not", they are both right according to grammaticality, but annotators usually come up with different corrections with different GEC standards.

5.4 Case Study

We select one case from CoNLL-2014 and visualize node interaction and node selection attention weights to study what VERNet learns from multi-hypotheses of beam search, as shown in Figure 5.

Given a source sentence, "Do one who suffered from this disease keep it a secret of infrom their relatives ?", and its five hypotheses from the Basic GEC Model, we plot the node interaction attention weights towards the word "suffers" in the hypothesis of node 2, which is assigned more higher score by BERT-VERNet. The word usage "suffers" is

Node1 **(17.41%)** Do one who suffered from this disease keep it a secret of in ##f ##rom their relatives ?
[SEP] <mark>Does</mark> someone who suffered from this disease keep it a secret from their relatives ? [SEP]
Node2 (24.09%) Do one who suffered from this disease keep it a secret of in ##f ##rom their relatives ?
[SEP] Does anyone who <mark>suffers</mark> from this disease keep it a secret from their relatives ? [SEP]
Node3 **(20.07%)** Do one who suffered from this disease keep it a secret of in ##f ##rom their relatives ?
[SEP] <mark>Does</mark> anyone who suffered from this disease keep it a secret from their relatives ? [SEP]
Node4 **(17.67%)** Do one who suffered from this disease keep it a secret of in ##f ##rom their relatives ?
[SEP] <mark>Does</mark> one who suffered from this disease keep it a secret from their relatives ? [SEP]
Node5 **(20.77%)** Do one who suffered from this disease keep it a secret of in ##f ##rom their relatives ?
[SEP] Does someone who <mark>suffers</mark> from this disease keep it a secret from their relatives ? [SEP]

Figure 5: Visualization of Attention Weight. Each node is the concatenation of the source sentence (with [SEP]) and a corresponding hypothesis sentence (with [SEP]). The selected node by BERT-VERNet is annotated (**Node2**). The node selection attention assigned to each hypothesis is annotated with dark orange. The node interaction attention towards the edited token "**suffers**" in the second node is also plotted. Darker red indicates higher attention weights.

more appropriate than "suffered" according to the context.

The node interaction attention accurately picks up the associated tokens "Does" from nodes 1, 3, and 4, and "suffers" from node 5. "Does" and "suffers" indicate the present tense and provide sufficient evidence to verify the quality of "suffers" in node 2. For node selection attention, the hypothesis (node 2) shares more attention than other nodes, which is more appropriate than other hypotheses. It demonstrates that the node attention is effective to select high-quality corrections with the source-hypothesis interactions.

The attention patterns are intuitive and effective, which further demonstrates VERNet's ability to well model the interactions of multi-hypotheses for better quality estimation.

6 Conclusion and Future Work

This paper presents VERNet for GEC quality estimation with multi-hypotheses. VERNet models the interactions of multiple hypotheses by building a reasoning graph, and then extracts clues with two kinds of attention: *node selection attention* and *node interaction attention*. They summarize and aggregate GEC evidence from multi-hypotheses to verify the quality of tokens. Experiments on four datasets show that VERNet achieves the state-of-the-art GED and quality estimation performance, and improves one published state-of-the-art GEC system. In the future, we will explore the impact of different kinds of hypotheses used in VERNet.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of EMNLP*, pages 4260–4270.

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

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of ACL*, pages 793–805.

Shamil Chollampatt and Hwee Tou Ng. 2018a. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of AAAI*, pages 5755–5762.

Shamil Chollampatt and Hwee Tou Ng. 2018b. Neural quality estimation of grammatical error correction. In *Proceedings of EMNLP*, pages 2528–2539.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of ACL*, pages 435–445.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of ACL*, pages 593–602.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of NAACL-HLT*, pages 568–572.

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 the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING*, pages 825–835.

Marina Fomicheva, Lucia Specia, and Francisco Guzmán. 2020. Multi-hypothesis machine translation evaluation. In *Proceedings of ACL*, pages 1218–1232.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of ACL*, pages 1055–1065.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Masato Hagiwara and Masato Mita. 2020. GitHub typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6761–6768.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an SMT approach to grammatical error correction. In *Proceedings of IJCAI*, pages 2803–2809.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of NAACL-HLT*, pages 595–606.

Masahiro Kaneko, Kengo Hotate, Satoru Katsumata, and Mamoru Komachi. 2019. TMU transformer system using BERT for re-ranking at BEA 2019 grammatical error correction on restricted track. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–212.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-head multi-layer attention to deep language representations for grammatical error detection. *Computación y Sistemas*, (3).

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 ACL*, pages 4248–4254.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of EMNLP*, pages 1236–1242.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of NAACL-HLT*, pages 3291–3301.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of EMNLP*, pages 5054–5065.

Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval*.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of IJCNLP*, pages 147–155.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of ACL*, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of EACL*, pages 229–234.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *Proceedings of ICML*, pages 3953–3962.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of ACL*, pages 2121–2130.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of AAAI*, pages 6916–6923.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Liang Wang, Wei Zhao, Ruoyu Jia, Sujian Li, and Jingming Liu. 2019a. Denoising based sequence-to-sequence pre-training for text generation. In *Proceedings of EMNLP*, pages 4003–4015.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019b. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of EMNLP*, pages 791–802.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*, pages 38–45.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of NAACL-HLT*, pages 619–628.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of ACL*, pages 180–189.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural sequence-labelling models for grammatical error correction. In *Proceedings of EMNLP*, pages 2795–2806.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of NAACL-HLT*, pages 380–386.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of NAACL-HLT*, pages 156–165.

A Appendices

A.1 Model Details of Sentence Quality Estimation Score Calculation

This part describes the details of sentence score calculation of BERT based quality estimation models.

Given a source sentence s with m tokens and k-th hypothesis c^k with n tokens, we can get the representation H^k of the k-th $\langle$source, hypothesis$\rangle$ sentence pair through BERT:

$$H^k = \text{BERT}([\text{CLS}] \, s \, [\text{SEP}] \, c^k \, [\text{SEP}]), \qquad (14)$$

or only the representation $\mathcal{H}^k$ of the k-th hypothesis through BERT:

$$\mathcal{H}^k = \text{BERT}([\text{CLS}] \, c^k \, [\text{SEP}]). \qquad (15)$$

The "[CLS]" representations are H_0^k and $\mathcal{H}_0^k$.

BERT-LM. We mask tokens in the k-th hypothesis sentence c^k and calculate the Perplexity of the k-th hypothesis sentence:

$$f_{\text{LM}}(c^k) = -\text{PPL}(\mathcal{H}_{1:n}^k). \qquad (16)$$

BERT-GQE. BERT-GQE uses the "[CLS]" representation $\mathcal{H}_0^k$ of k-th hypothesis to estimate the sentence quality with the probability $P(y_s|c^k)$:

$$P(y_s|c^k) = \text{softmax}_{y_s}(W \cdot \mathcal{H}_0^k), \qquad (17)$$

where W is the parameter and the label y_s is categorized into two groups: correct ($y_s = 1$) and incorrect ($y_s = 0$).

Then the sentence-level quality estimation score of hypothesis c^k is calculated:

$$f_{\text{GQE}}(c^k) = P(y_s = 1|c^k). \qquad (18)$$

BERT-QE. BERT-QE uses the "[CLS]" representation H_0^k of k-th $\langle$source, hypothesis$\rangle$ sentence pair to estimate the quality of GEC hypothesis:

$$f_{\text{QE}}(s, c^k) = \text{sigmoid}(W \cdot H_0^k), \qquad (19)$$

where W is the parameter. The quality estimation score $f_{\text{QE}}(s, c^k)$ of BERT-QE is trained to approximate the $F_{0.5}$ score of the k-th hypothesis c^k.

BERT-GED. Take BERT-GED (HYP) as an example, it uses the hypothesis representation $H_{m+2:m+n+2}^k$ of the k-th $\langle$source, hypothesis$\rangle$ sentence pair to estimate the quality of GEC hypothesis. Note that the "[SEP]" token is also used in BERT-GED to denote the end of the sentence.

Model	P	R	$F_{0.5}$
LSTM	58.88	28.92	48.48
BiLSTM-ATTN	60.73	22.33	45.07
BiLSTM-JOINT	65.53	28.61	52.07
BERT	**73.69**	**45.39**	**65.52**

Table 6: Grammatical Error Detection Performance on the First Certificate in English (FCE) dataset (Yannakoudakis et al., 2011).

We calculate the probability of token quality estimation label y for the i-th token w_i^k in the k-th $\langle$source, hypothesis$\rangle$ sentence pair:

$$P(y|w_i^k) = \text{softmax}(W \cdot H_i^k), \qquad (20)$$

where W is the parameter. The label y is categorized into two groups: correct ($y = 1$) and incorrect ($y = 0$).

To estimate the quality of hypotheses, we average all token quality estimation probability $P(y = 1|w_i^k)$ as the sentence quality estimation score $f(s, c^k)$ for the k-th hypothesis c^k:

$$f_{\text{GED}}(s, c^k) = \frac{1}{n+1} \sum_{i=m+2}^{m+n+2} P(y = 1|w_i^k). \qquad (21)$$

A.2 Grammatical Error Detection Performance with LSTM

In this experiment, we evaluate the effectiveness of BERT and LSTM on the grammatical error detection (GED) task. We keep the same setting as previous work (Rei and Søgaard, 2019). The FCE dataset is used for evaluation. Precision, Recall, and $F_{0.5}$ are used as our evaluation metrics.

As shown in Table 6, three models, LSTM, LSTM-ATTN, and LSTM-JOINT from Rei and Søgaard (2019) are compared with the BERT model. The LSTM model leverages the LSTM encoder and adds language modeling objectives in the training process (Rei, 2017). LSTM-ATTN and LSTM-JOINT further add attention constraints and sentence level supervision to achieve better performance (Rei and Søgaard, 2019). The BERT model is the same as our BERT-GED (SRC).

The BERT based model shows significant improvement than LSTM based models. Thus we do not consider LSTM based GED models in the experiments of GEC quality estimation.