Neural and FST-based approaches to grammatical error correction

Zheng Yuan[♠◇] **Felix Stahlberg**[♣] **Marek Rei**[♠◇] **Bill Byrne**[♣] **Helen Yannakoudakis**[♠◇]
[♠]Department of Computer Science & Technology, University of Cambridge, United Kingdom
[◇]ALTA Institute, University of Cambridge, United Kingdom
[♣]Department of Engineering, University of Cambridge, United Kingdom
{zheng.yuan, marek.rei, helen.yannakoudakis}@cl.cam.ac.uk,
{fs439, bill.byrne}@eng.cam.ac.uk

Abstract

In this paper, we describe our submission to the BEA 2019 shared task on grammatical error correction. We present a system pipeline that utilises both error detection and correction models. The input text is first corrected by two complementary neural machine translation systems: one using convolutional networks and multi-task learning, and another using a neural Transformer-based system. Training is performed on publicly available data, along with artificial examples generated through back-translation. The n-best lists of these two machine translation systems are then combined and scored using a finite state transducer (FST). Finally, an unsupervised re-ranking system is applied to the n-best output of the FST. The re-ranker uses a number of error detection features to re-rank the FST n-best list and identify the final 1-best correction hypothesis. Our system achieves 66.75% $F_{0.5}$ on error correction (ranking 4th), and 82.52% $F_{0.5}$ on token-level error detection (ranking 2nd) in the restricted track of the shared task.

1 Introduction

Grammatical error correction (GEC) is the task of automatically correcting grammatical errors in written text. In this paper, we describe our submission to the restricted track of the BEA 2019 shared task on grammatical error correction (Bryant et al., 2019), where participating teams are constrained to using only the provided datasets as training data. Systems are expected to correct errors of all types, including grammatical, lexical and orthographical errors. Compared to previous shared tasks on GEC, which have primarily focused on correcting errors committed by non-native speakers (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014), a new annotated dataset is introduced, consisting of essays produced by native and non-native English language learners, with a wide coverage of language proficiency levels for the latter, ranging from elementary to advanced.

Neural machine translation (NMT) systems for GEC have drawn growing attention in recent years (Yuan and Briscoe, 2016; Xie et al., 2016; Ji et al., 2017; Sakaguchi et al., 2017; Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018), as they have been shown to achieve state-of-the-art results (Ge et al., 2018; Zhao et al., 2019). Within this framework, error correction is cast as a monolingual translation task, where the source is a sentence (written by a language learner) that may contain errors, and the target is its corrected counterpart in the same language.

Due to the fundamental differences between a "true" machine translation task and the error correction task, previous work has investigated the adaptation of NMT for the task of GEC. Byte pair encoding (BPE) (Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018) and a copying mechanism (Zhao et al., 2019) have been introduced to deal with the "noisy" input text in GEC and the non-standard language used by learners. Some researchers have investigated ways of incorporating task-specific knowledge, either by directly modifying the training objectives (Schmaltz et al., 2017; Sakaguchi et al., 2017; Junczys-Dowmunt et al., 2018) or by re-ranking machine-translation-system correction hypotheses (Yannakoudakis et al., 2017; Chollampatt and Ng, 2018). To ameliorate the lack of large amounts of error-annotated learner data, various approaches have proposed to leverage unlabelled native data within a number of frameworks, including artificial error generation with back translation (Rei et al., 2017; Kasewa et al., 2018), fluency boost learning (Ge et al., 2018), and pre-training with denoising autoencoders (Zhao et al., 2019).

Previous work has shown that a GEC system

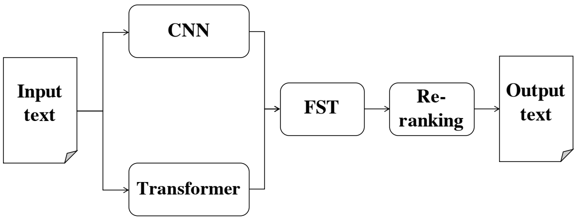

Figure 1: Overview of our best GEC system pipeline.

targeting all errors may not necessarily be the best approach to the task, and that different GEC systems may be better suited to correcting different types of errors, and can therefore be complementary (Yuan, 2017). As such, hybrid systems that combine different approaches have been shown to yield improved performance (Felice et al., 2014; Rozovskaya and Roth, 2016; Grundkiewicz and Junczys-Dowmunt, 2018). In line with this work, we present a hybrid approach that 1) employs two NMT-based error correction systems: a neural convolutional system and a neural Transformer-based system; 2) a finite state transducer (FST) that combines and further enriches the n-best outputs of the NMT systems; 3) a re-ranking system that re-ranks the n-best output of the FST based on error detection features.

The remainder of this paper is organised as follows: Section 2 describes our approach to the task; Section 3 describes the datasets used and presents our results on the shared task development set; Section 4 presents our official results on the shared task test set, including a detailed analysis of the performance of our final system; and, finally, Section 5 concludes the paper and provides an overview of our findings.

2 Approach

We approach the error correction task using a pipeline of systems, as presented in Figure 1. In the following sections, we describe each of these components in detail.

2.1 The convolutional neural network (CNN) system

We use a neural sequence-to-sequence model and an encoder–decoder architecture (Cho et al., 2014; Sutskever et al., 2014). An encoder first reads and encodes an entire input sequence $\mathbf{x} = (x_1, x_2, ..., x_n)$ into hidden state representations. A decoder then generates an output sequence $\mathbf{y} = (y_1, y_2, ..., y_m)$ by predicting the next token y_t based on the input sequence $\mathbf{x}$ and all the previously generated tokens $\{y_1, y_2, ..., y_{t-1}\}$:

$$p(\mathbf{y}) = \prod_{t=1}^{m} p(y_t|\{y_1, ..., y_{t-1}\}, \mathbf{x}) \qquad (1)$$

Our convolutional neural system is based on a multi-layer convolutional encoder–decoder model (Gehring et al., 2017), which employs convolutional neural networks (CNNs) to compute intermediate encoder and decoder states. The parameter settings follow Chollampatt and Ng (2018) and Ge et al. (2018). The source and target word embeddings have size 500, and are initialised with fastText embeddings (Bojanowski et al., 2017) trained on the native English Wikipedia corpus (2,405,972,890 tokens). Each of the encoder and decoder is made up of seven convolutional layers, with a convolution window width of 3. We apply a left-to-right beam search to find a correction that approximately maximises the conditional probability in Equation 1.

BPE is introduced to alleviate the rare-word problem, and rare and unknown words are split into multiple frequent subword tokens (Sennrich et al., 2016b). NMT systems often limit vocabulary size on both source and target sides due to the computational complexity during training. Therefore, they are unable to translate out-of-vocabulary (OOV) words, which are treated as unknown tokens, resulting in poor translation quality. As noted by Yuan and Briscoe (2016), this problem is more serious for GEC as non-native text contains, not only rare words (e.g., proper nouns), but also misspelled words (i.e., spelling errors).

In our model, each of the source and target vocabularies consist of the 30K most frequent BPE tokens from the source and target side of the parallel training data respectively. The same BPE operation is applied to the Wikipedia data before being used for training of our word embeddings.

Copying mechanism is a technique that has led to performance improvement on various monolingual sequence-to-sequence tasks, such as text summarisation, dialogue systems, and paraphrase generation (Gu et al., 2016; Cao et al., 2017). The idea is to allow the decoder to choose between simply copying an original input word and outputting a translation word. Since the source and target sentences are both in the same language (i.e., monolingual translation) and most words in the source sentence are correct and do not need to change, GEC seems to benefit from the copying mechanism.

Following the work of Gu et al. (2016), we use a dynamic target vocabulary, which contains a fixed vocabulary learned from the target side of the training data plus all the unique tokens introduced by the source sentence. As a result, the probability of generating any target token $p(y_t|\{y_1, ..., y_{t-1}\}, \mathbf{x})$ in Equation 1 is defined as a "mixture" of the generation probability $p(y_t, g|\{y_1, ..., y_{t-1}\}, \mathbf{x})$ and the copy probability $p(y_t, c|\{y_1, ..., y_{t-1}\}, \mathbf{x})$:

$$p(y_t|\{y_1, ..., y_{t-1}\}, \mathbf{x}) = p(y_t, g|\{y_1, ..., y_{t-1}\}, \mathbf{x}) + p(y_t, c|\{y_1, ..., y_{t-1}\}, \mathbf{x}) \quad (2)$$

Multi-task learning has found success in a wide range of tasks, from natural language processing (NLP) (Collobert and Weston, 2008) and speech recognition (Deng et al., 2013) to computer vision (Girshick, 2015). Multi-task learning allows systems to use information from related tasks and learn from multiple objectives, which leads to performance improvement on individual tasks. Recently, Rei (2017) and Rei and Yannakoudakis (2017) investigated the use of different auxiliary objectives for the task of error detection in learner writing.

In addition to our primary error correction task, we propose two related auxiliary objectives to boost model performance:

- **Token-level labelling**

  We jointly train an error detection and error correction system by providing error detection labels. Instead of only generating a corrected sentence, we extend the system to additionally predict whether a token in the source sentence is correct or incorrect.

- **Sentence-level labelling**

  A binary classification task is also introduced to predict whether the original source sentence is grammatically correct or incorrect. We investigate the usefulness of sentence-level classification as an auxiliary objective for training error correction models.

Labels for both auxiliary error detection tasks are generated automatically by comparing source and target tokens using the ERRANT automatic alignment tool (Bryant et al., 2017). We first align each token x_i in the source sentence $\mathbf{x}$ with a token y_j in the target sentence $\mathbf{y}$. If $x_i = y_j$, the source token x_i is correct; while if $x_i \neq y_j$, the source token x_i is incorrect. Similarly, the source sentence $\mathbf{x}$ is correct if $\mathbf{x} = \mathbf{y}$, and incorrect otherwise.

Artificial error generation is the process of injecting artificial errors into a set of error-free sentences. Compared to standard machine translation tasks, GEC suffers from the limited availability of large amounts of training data. As manual error annotation of learner data is a slow and expensive process, artificial error generation has been applied to error correction (Felice and Yuan, 2014) and detection (Rei et al., 2017) with some success. Following the work of Rei et al. (2017), we treat error generation as a machine translation task, where a grammatically correct sentence is translated to an incorrect counterpart. We built an error generation system using the same network architecture as the one described here, with error-corrected sentences as the source and their corresponding uncorrected counterparts written by language learners as the target. The system is then used to collect the n-best outputs: $\mathbf{y}_o^1, \mathbf{y}_o^2, ..., \mathbf{y}_o^n$, for a given error-free native and/or learner sentence $\mathbf{y}$. Since there is no guarantee that the error generation system will inject errors into the input sentence $\mathbf{y}$ to make it less grammatically correct, we apply "quality control". A pair of artificially generated sentences $(\mathbf{y}_o^k, \mathbf{y})$, for $k \in \{1, 2, ..., n\}$, will be added to the training set of the error correction system if the following condition is met:

$$\frac{f(\mathbf{y})}{f(\mathbf{y}_o^k)} \leq \sigma \quad (3)$$

where $f(\mathbf{y})$ is the normalised log probability of $\mathbf{y}$:

$$f(\mathbf{y}) = \frac{\sum_{t=1}^{m} \log(P(y_t|\mathbf{y}_{<t}))}{m} \quad (4)$$

This ensures that the quality of the artificially generated sentence, as estimated by a language model, is lower compared to the original sentence. We use a 5-gram language model (LM) trained on the One Billion Word Benchmark dataset (Chelba et al., 2014) with KenLM (Heafield, 2011) to compute $P(y_t|\mathbf{y}_{<t})$.

The σ in Equation 3 is a threshold used to filter out sentence pairs with unnecessary changes; e.g., [*I look forward to hearing from you.* → *I am looking forward to hearing from you.*]. It is an averaged score learned on the development set:

$$\sigma = \frac{\sum_{i=1}^{N} \frac{f(\mathbf{y}_i)}{f(\mathbf{x}_i)}}{N} \tag{5}$$

where $(\mathbf{x}, \mathbf{y})$ is a pair of parallel sentences in the development set, and N is the total number of pairs.

2.2 The neural Transformer-based system

Besides the convolutional system from the previous section, we also use the purely neural Transformer-based system of Stahlberg and Byrne (2019). They use an ensemble of four Transformer (Vaswani et al., 2017) NMT and two Transformer LM models in Tensor2Tensor (Vaswani et al., 2018) `transformer_big` configuration. The NMT models are trained with back-translation (Sennrich et al., 2016a; Rei et al., 2017; Kasewa et al., 2018) and fine-tuning through continued training. For a detailed description of this system we refer the reader to Stahlberg and Byrne (2019).

2.3 FST-based system combination

Stahlberg et al. (2019) demonstrated the usefulness of FSTs for grammatical error correction. Their method starts with an input lattice I which is generated with a phrase-based statistical machine translation (SMT) system. The lattice I is composed with a number of FSTs that aim to enrich the search space with further possible corrections. Similarly to Bryant and Briscoe (2018), they rely on external knowledge sources like spell checkers and morphological databases to generate additional correction options for the input sentence. The enriched lattice is then mapped to the sub-word level by composition with a mapping transducer, and re-scored with neural machine translation models and neural LMs.

In this work, rather than combining SMT and neural models, we use the framework of Stahlberg et al. (2019) to combine and enrich the outputs of two neural systems. The input lattice I is now the union of two n-best lists – one from the convolutional system (Section 2.1), and one from the Transformer-based system (Section 2.2). After composition, we re-score the enriched input lattice I with the system described in Section 2.2. The FST-based system combination uses 7 different features: the convolutional system score, the LM and NMT scores from the Transformer-based system, the edit distance of hypotheses in I to the input sentence, substitution and deletion penalties for the additional correction options from the FST framework, and the word count. Following Stahlberg et al. (2019); Stahlberg and Byrne (2019), we scale these features and tune the scaling weights on the BEA-2019 development set using a variant of Powell search (Powell, 1964). We use OpenFST (Allauzen et al., 2007) as backend for FST operations, and the SGNMT decoder (Stahlberg et al., 2017, 2018) for neural decoding under FST constraints.

2.4 Re-ranking FST output

Yannakoudakis et al. (2017) found that grammatical error detection systems can be used to improve error correction outputs. Specifically, they re-rank the n-best correction hypotheses of an SMT system based on error detection predictions. Following this work, we also deploy a re-ranking component which re-ranks the n-best correction hypotheses of the FST system (Section 2.3) based on error detection predictions output by an error detection system.

Error detection. Our system for grammatical error detection is based on the model described by Rei (2017).[1] The task is formulated as a sequence labeling problem – given a sentence, the model assigns a probability to each token, indicating the likelihood of that token being incorrect in the given context (Rei and Yannakoudakis, 2016). The architecture maps words to distributed embeddings, while also constructing character-based representations for each word with a neural component. These are then passed through a bidirectional LSTM, followed by a feed-forward layer and a softmax layer at the output.

In addition to neural text representations, we also include several external features into the model, designed to help it learn more accurate error detection patterns from the limited amounts of training data available:

- Two binary features indicating whether two publicly available spell-checkers – HunSpell[2] and JamSpell[3] – identify the target word as a spelling mistake.

[1] `https://github.com/marekrei/sequence-labeler`

[2] `http://hunspell.github.io/`

[3] `https://github.com/bakwc/JamSpell`

- The POS tag, NER label and dependency relation of the target word based on the Stanford parser (Chen and Manning, 2014).
- The number of times the unigram, bigram, or trigram context of the target word appears in the BNC (Burnard, 2007) and in ukWaC (Ferraresi et al., 2008).
- Contextualized word representations from ELMo (Peters et al., 2018).

The discrete features are represented as 10-dimensional embeddings and, together with the continuous features, concatenated to each word representation in the model. The overall architecture is optimized for error detection using cross-entropy. Once trained, the model returns the predicted probabilities of each token in a sentence being correct or incorrect.

Re-ranker. We generate the list of the 8 best correction hypotheses from our FST system, and then use the following set of error detection-based features to assign a new score to each hypothesis and determine a new ranking:

1. Sentence correctness probability: the error detection model outputs a probability indicating whether a token is likely to be correct or incorrect in context. We therefore use as a feature the overall FST sentence probability, calculated based on the probability of each of its tokens being correct: $\sum_w \log P(w)$
2. Levenshtein distance (LD): we first use LD to identify 1) which tokens in the original/uncorrected sentence have been corrected by the FST candidate hypothesis, and 2) which tokens in the original/uncorrected sentence our detection model predicts as incorrect (i.e., the probability of being incorrect is > 0.5). We then convert these annotations to binary sequences – i.e., 1 if the token is identified as incorrect, and 0 otherwise – and use as a feature the LD between those binary representations. Specifically, we would like to select the candidate FST sentence that has the smallest LD from the binary sequence created by the detection model, and therefore use as a feature the following: $\frac{1.0}{\text{LD}+1.0}$
3. False positives: using the binary sequences described above, we count the number of false positives (FP) on token-level error detection by treating the error detection model as the "gold standard". Specifically, we count how many times the candidate FST hypothesis disagrees with the detection model on the tokens identified as incorrect, and use as a feature the following: $\frac{1.0}{\text{FP}+1.0}$

We use a linear combination of the above three features together with the original score given by the FST system for each candidate hypothesis to re-rank the FST system's 8-best list in an unsupervised way. The new 1-best correction hypothesis c^* is then the one that maximises:

$$c^* = \arg\max_c \sum_{i=1}^{K} \lambda_i \, h_i(c) \qquad (6)$$

where h represents the score assigned to candidate hypothesis c according to feature i; λ is a weighting parameter that controls the effect feature i has on the final ranking; and $K = 4$ as we use a total of four different features (three features based on the detection model, and one which is the original score output by the FST system). λs are tuned on the development set and are set to $\lambda = 2.0$ for features 1. and 2., $\lambda = 3.0$ for feature 3. and $\lambda = 1.5$ for the original FST score.

3 Experiments and results

3.1 Datasets and evaluation

In the restricted track, participating teams were constrained to use only the provided learner datasets:[4]

- **Cambridge English W&I corpus**

  Cambridge English Write & Improve (W&I)[5] (Yannakoudakis et al., 2018) is an online web platform that assists non-native English learners with their writing. Learners from around the world submit letters, stories, articles and essays for automated assessment in response to various prompts. The W&I corpus (Bryant et al., 2019) contains $3,600$ annotated submissions across 3 different CEFR[6] levels: A (beginner), B (intermediate), and C (advanced). The data has been

[4] We note that there are no restrictions on the use of NLP tools (e.g., POS taggers, parsers, spellcheckers, etc.), nor on the amount of unannotated data that can be used, so long as such resources are publicly available.

[5] https://writeandimprove.com/

[6] https://www.cambridgeenglish.org/exams-and-tests/cefr/

split into training ($3,000$ essays), development (200 essays), and test (200 essays) partitions.

- **LOCNESS**

  The LOCNESS[7] corpus (Granger, 1998) consists of essays written by native English students. A subsection of 100 essays has been manually annotated, and equally partitioned into development and test sets.

- **FCE**

  The First Certificate in English (FCE) corpus (Yannakoudakis et al., 2011) is a subset of the Cambridge Learner Corpus (CLC) that consists of $1,244$ exam scripts written by learners of English sitting the FCE exam.

- **NUCLE**

  The National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) contains $1,400$ essays written by undergraduate students at the National University of Singapore who are non-native English speakers.

- **Lang-8 Corpus of Learner English**

  Lang-8[8] is an online language learning website which encourages users to correct each other's grammar. The Lang-8 Corpus of Learner English (Mizumoto et al., 2011; Tajiri et al., 2012) refers to an English subsection of this website (can be quite noisy).

Additional resources used in our system include:

- **English Wikipedia corpus**

  The English Wikipedia corpus ($2,405,972,890$ tokens in $110,698,467$ sentences) is used to pre-train word embeddings for the convolutional neural system. We also use it as error-free native data for artificial error generation (see Section 2.1).

- **One Billion Word Benchmark dataset**

  A LM is trained on the One Billion Word Benchmark dataset, which consists of close to a billion words of English taken from news articles on the web, to evaluate the quality of artificially generated sentence pairs. A filtered version ($768,646,526$ tokens in $30,301,028$ sentences) is used as input to the error generation model in Section 2.1.

In order to cover the full range of English levels and abilities, the official development set consists of 300 essays from W&I (A: 130, B:100, and C:70) and 50 essays from LOCNESS ($86,973$ tokens in $4,384$ sentences).

The ERRANT scorer (Bryant et al., 2017) is used as the official scorer for the shared task. System performance is evaluated in terms of span-level correction using $\mathrm{F}_{0.5}$, which emphasises precision twice as much as recall.

3.2 Training details

The convolutional NMT model is trained with a hidden layer size of $1,024$ for both the encoder and the decoder. Dropout at a rate of 0.2 is applied to the embedding layers, convolutional layers and decoder output. The model is optimized using Nesterov's Accelerated Gradient Descent (NAG) with a simplified formulation for Nesterov's momentum (Bengio et al., 2013). The initial learning rate is set to 0.25, with a decaying factor of 0.1 and a momentum value of 0.99. We perform validation after every epoch, and select the best model based on the performance on the development set. During beam search, we keep a beam size of 12 and discard all other hypotheses.

The grammatical error detection system was optimized separately as a sequence labeling model. Word embeddings were set to size 300 and initialized with pre-trained Glove embedding (Pennington et al., 2014). The bi-LSTM has 300-dimensional hidden layers for each direction. Dropout was applied to word embeddings and LSTM outputs with probability 0.5. The model was optimized with Adam (Kingma and Ba, 2015), using a default learning rate 0.001. Training was stopped when performance on the development set did not improved over 7 epochs.

3.3 Individual system performance

Individual system performance on the development set is reported in Table 1, where 'CNN' refers to the convolutional neural system, and 'Transformer' refers to the Transformer-based neural system. These results are based on the 1-best output from each system, although the n-best

[7] https://uclouvain.be/en/research-institutes/ilc/cecl/locness.html

[8] https://lang-8.com/

System	TP	FP	FN	P	R	$F_{0.5}$
CNN	1697	2371	5858	41.72	22.46	35.61
Transformer	2455	2162	5006	53.17	32.90	47.34

Table 1: Span-level correction results for individual systems on the development set. TP: true positives, FP: false positives, FN: false negatives, P: precision, R: recall.

lists are used later for system combination.

3.4 Pipelines

Since corrections made by the convolutional neural system and the Transformer-based system are often complementary, and re-scoring has been proven to be useful and effective for error correction, we investigated ways to combine corrections generated by both systems. Table 2 shows results for different combinations, where 'CNN' refers to the convolutional neural system, 'Transformer' refers to the Transformer-based system, subscript '10-best' indicates the use of the 10-best list of correction candidates from the system, '+' indicates a combination of corrections from different systems, and '>' indicates a pipeline where the output of one system is the input to the other.

4 Official evaluation results

Our submission to the shared task is the result of our best hybrid system, described in Section 2 and summarised in Figure 1. Similar to the official development set, the test set comprises 350 texts (85,668 tokens in 4,477 sentences) written by native and non-native English learners.

Systems were evaluated using the ERRANT scorer, with span-based correction $F_{0.5}$ as the primary measure. In the restricted track, where participants were constrained to use only the provided training sets, our submitted system ranked fourth[9] out of 21 participating teams. The official results of our submission in terms of span-level correction, span-level detection and token-level detection, including our system rankings, are reported in Table 3. It is worth noting that our correction system yielded particularly high performance on error detection tasks, ranking third on span-level detection and second on token-level detection. We believe that much of the success in error detection can be credited to the error detection auxiliary objectives introduced in the convolutional neural system (see Section 2.1) and the error detection features used in our final re-ranking system (see Section 2.4).

We also report span-level correction performance in terms of different CEFR levels (A, B, and C),[10] as well as on the native texts only (N) in Table 4. Our final error correction system performs best on advanced learner data (C), achieving an $F_{0.5}$ score of 73.28, followed by intermediate learner data (B), native data (N), and lastly beginner learner data (A). The difference between the highest and lowest $F_{0.5}$ scores is 8.12 points. We also note that the system seems to be handling errors made by native students effectively even though it has not been trained on any native parallel data. Overall, we observe that our system generalises well across native and non-native data, as well as across different proficiency/CEFR levels.

In order to better understand the performance of our hybrid error correction system, we perform a detailed error analysis. This helps us understand the strengths and weaknesses of the system, as well as identify areas for future work. Error type-specific performance is presented in Table 5. We can see that our system achieves the highest results on *VERB:INFL* (verb inflection) errors with an $F_{0.5}$ of 93.75. However, the result is not truly representative as there are only 8 verb inflection errors in the test data, and our system successfully corrects 6 of them. The error type that follows is *ORTH* (orthography), which comprises case and/or whitespace errors. A high precision score of 89.11 is observed, suggesting that our system is particularly suitable for these kind of errors. We also observe that our system is effective at correcting *VERB:SVA* (subject–verb agreement) errors, achieving an $F_{0.5}$ of 80.08. Results for *ADJ:FORM* (adjective form; $F_{0.5}$=78.95) and *CONTR* (contraction; $F_{0.5}$=77.92) are high; however, these error types only account for small fractions of the test set (0.188% and 0.245% respectively).

The worst performance is observed for type *CONJ* (conjunction), with an $F_{0.5}$ of 28.46. Our system successfully corrected 7 conjunction errors, while missed 20 and made 17 unnecessary changes. We note that our system is less effective at correcting open-class errors

[9] The system is tied for third place as the difference in $F_{0.5}$ is negligible.

[10] https://www.cambridgeenglish.org/exams-and-tests/cefr/

Pipeline	TP	FP	FN	P	R	$F_{0.5}$
$\text{CNN}_{\text{10-best}} + \text{Transformer}_{\text{10-best}} > \text{FST}$	2416	1798	5045	57.33	32.38	49.68
$\text{CNN}_{\text{10-best}} + \text{Transformer}_{\text{10-best}} > \text{FST}_{\text{8-best}} > \text{Re-ranking}$	2502	1839	4959	57.64	33.53	50.39

Table 2: Span-level correction results for different system pipelines on the development set.

Evaluation	TP	FP	FN	P	R	$F_{0.5}$	#
Span-level correction	2924	1224	2386	70.49	55.07	66.75	4
Span-level detection	3383	774	2181	81.38	60.80	76.22	3
Token-level detection	4098	470	2461	89.71	62.48	82.52	2

Table 3: Official results of our submitted system on the test set.

Level	TP	FP	FN	P	R	$F_{0.5}$
A	1272	573	1108	68.94	53.45	65.16
B	905	368	806	71.09	52.89	66.51
C	425	131	251	76.44	62.87	73.28
N	322	152	221	67.93	59.30	66.01

Table 4: Proficiency level-specific span-level correction performance of our submitted system on the test set. A: CEFR beginner; B: CEFR intermediate; C: CEFR advanced; N: native.

such as *NOUN* (noun; $F_{0.5}$=34.75), *OTHER* (other;[11] $F_{0.5}$=38.95); *VERB* (verb; $F_{0.5}$=39.80); *ADJ* (adjective; $F_{0.5}$=41.94) and *ADV* (adverb; $F_{0.5}$=51.65) errors. As noted by Kochmar (2016), such error types are quite challenging for error detection and correction systems: content words express meaning, and their semantic properties should be taken into account. Unlike errors in function words, content word errors are often less systematic; e.g., [*person* → *people*, *ambulate* → *walk*, *big* → *wide*, *speedily* → *quickly*].

5 Conclusion

In this paper, we have presented a hybrid approach to error correction that combines a convolutional and a Transformer-based neural system. We have explored different combination techniques involving sequential pipelines, candidate generation and re-ranking. Our best hybrid system submitted to the restricted track of the BEA 2019 shared task yields a span-level correction score of $F_{0.5} = 66.75$, placing our system in the fourth place out of 21 participating teams. High results were observed for both span-level and token-level error detection (ranking our system third and second respectively), suggesting that our error correction system can also effectively detect errors. Detailed analyses show that our system generalises well across different language proficiency levels (CEFR) and native / non-native domains. An error-type analysis showed that our system is particularly good at correcting verb inflection, orthography and subject–verb agreement errors, but less effective at correcting open-class word errors which are less systematic.

Error type	TP	FP	FN	P	R	$F_{0.5}$
ADJ	13	15	30	46.43	30.23	41.94
ADJ:FORM	6	1	4	85.71	60.00	78.95
ADV	25	19	41	56.82	37.88	51.65
CONJ	7	17	20	29.17	25.93	28.46
CONTR	12	4	1	75.00	92.31	77.92
DET	421	149	228	73.86	64.87	71.87
MORPH	91	20	60	81.98	60.26	76.47
NOUN	36	63	86	36.36	29.51	34.75
NOUN:INFL	7	1	13	87.50	35.00	67.31
NOUN:NUM	199	85	64	70.07	75.67	71.12
NOUN:POSS	29	10	25	74.36	53.70	69.05
ORTH	229	28	162	89.11	58.57	80.69
OTHER	160	181	530	46.92	23.19	38.95
PART	24	8	10	75.00	70.59	74.07
PREP	262	125	193	67.70	57.58	65.40
PRON	59	20	82	74.68	41.84	64.55
PUNCT	636	192	291	76.81	68.61	75.02
SPELL	204	47	108	81.27	65.38	77.51
VERB	57	64	175	47.11	24.57	39.80
VERB:FORM	157	52	45	75.12	77.72	75.63
VERB:INFL	6	0	2	100.00	75.00	93.75
VERB:SVA	127	32	30	79.87	80.89	80.08
VERB:TENSE	122	64	137	65.59	47.10	60.82
WO	35	27	49	56.45	41.67	52.71

Table 5: Error type-specific span-level correction performance of our submitted system on the test set.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. Zheng Yuan, Marek Rei and Helen Yannakoudakis were supported by Cambridge Assessment, University of Cambridge. Felix Stahlberg and Bill Byrne acknowledge support for this work by the U.K. Engineering and Physical Sciences Research Council (EPSRC) grant EP/L027623/1 and grant EP/P020259/1 which provided Cambridge Tier-2 resources operated by the University of Cambridge Research Computing

[11] Errors that do not fall into any other category (e.g., paraphrasing).

Service.[12] We thank the NVIDIA Corporation for the donation of the Titan X Pascal GPU used in this research.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer.

Yoshua Bengio, Nicolas Boulanger-lew, and Razvan Pascanu. 2013. Advances in optimizing recurrent networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8624–8628, Vancouver, BC, Canada. IEEE.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253. 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 the 14th Workshop on Innovative Use of NLP for Building Educational Applications*, 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 the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Lou Burnard. 2007. Reference Guide for the British National Corpus (XML Edition). Technical report.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3152–3158, San Francisco, California, USA. Association for the Advancement of Artificial Intelligence.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of INTERSPEECH 2014*, pages 2635–2639, Singapore. International Speech Communication Association.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5755–5762, New Orleans, Louisiana, USA. Association for the Advancement of Artificial Intelligence.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM.

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, Atlanta, Georgia. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: an overview. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603, Vancouver, BC, Canada. IEEE.

[12] http://www.hpc.cam.ac.uk

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. Association for Computational Linguistics.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. Technical report, Microsoft Research.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252, Sydney, Australia.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pages 1440–1448, Santiago, Chile. IEEE.

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.

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 the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Ekaterina Kochmar. 2016. Error detection in content word combinations. Technical Report UCAM-CL-TR-886, Computer Laboratory, University of Cambridge.

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 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

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, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe : Global Vectors for Word Representation. In *Proceedings of the Empiricial Meth- ods in Natural Language Processing (EMNLP 2014)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

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

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2016. Compositional Sequence Labeling Models for Error Detection in Learner Writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

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

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2205–2215, Berlin, Germany. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 366–372, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Felix Stahlberg, Christopher Bryant, and Bill Byrne. 2019. Neural grammatical error correction with finite state transducers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. The CUED's grammatical error correction systems for BEA19. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*, Florence, Italy. Association for Computational Linguistics.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT – A flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30. Association for Computational Linguistics.

Felix Stahlberg, Danielle Saunders, Gonzalo Iglesias, and Bill Byrne. 2018. Why not be versatile? Applications of the SGNMT decoder for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 208–216. Association for Machine Translation in the Americas.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob

Uszkoreit. 2018. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *arXiv*, abs/1603.09727.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31(3):251–267.

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

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806, Copenhagen, Denmark. Association for Computational Linguistics.

Zheng Yuan. 2017. Grammatical error correction in non-native english. Technical Report UCAM-CL-TR-904, Computer Laboratory, University of Cambridge.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.