# The BEA-2019 Shared Task on Grammatical Error Correction

**Christopher Bryant** **Mariano Felice** **Øistein E. Andersen** **Ted Briscoe**
ALTA Institute
Computer Laboratory
University of Cambridge
Cambridge, UK
{cjb255,mf501,oa223,ejb}@cam.ac.uk

## Abstract

This paper reports on the BEA-2019 Shared Task on Grammatical Error Correction (GEC). As with the CoNLL-2014 shared task, participants are required to correct all types of errors in test data. One of the main contributions of the BEA-2019 shared task is the introduction of a new dataset, the Write&Improve+LOCNESS corpus, which represents a wider range of native and learner English levels and abilities. Another contribution is the introduction of tracks, which control the amount of annotated data available to participants. Systems are evaluated in terms of ERRANT $F_{0.5}$, which allows us to report a much wider range of performance statistics. The competition was hosted on Codalab and remains open for further submissions on the blind test set.

## 1 Introduction

The Building Educational Applications (BEA) 2019 Shared Task on Grammatical Error Correction (GEC) continues the tradition of the previous Helping Our Own (HOO) and Conference on Natural Language Learning (CoNLL) shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014) and was motivated by the need to re-evaluate the field after a five year hiatus. Although significant progress has been made since the end of the last CoNLL-2014 shared task, recent systems have been trained, tuned and tested on different combinations of metrics and corpora (Sakaguchi et al., 2017; Yannakoudakis et al., 2017; Chollampatt and Ng, 2018a; Ge et al., 2018; Grundkiewicz and Junczys-Dowmunt, 2018; Junczys-Dowmunt et al., 2018; Lichtarge et al., 2018; Zhao et al., 2019). Thus one of the main aims of the BEA-2019 shared task is to once again provide a platform where systems can be re-evaluated under more controlled conditions.

With this in mind, another significant contribution of the BEA-2019 shared task is the introduction of a new annotated dataset, the Cambridge English Write & Improve (W&I) and LOCNESS corpus, which is designed to represent a much wider range of English levels and abilities than previous corpora. This is significant because systems have traditionally only been tested on the CoNLL-2014 test set, which only contains 50 essays (1,312 sentences) on 2 different topics written by 25 South-East Asian undergraduates (Ng et al., 2014). In contrast, the W&I+LOCNESS test set contains 350 essays (4,477 sentences) on approximately 50 topics written by 334 authors from around the world (including native English speakers). We hope that this diversity will encourage the development of systems that can generalise better to unseen data.

Another difference to the previous shared tasks is the introduction of tracks; namely the Restricted, Unrestricted and Low Resource track. While annotated data was comparatively scarce five years ago, it has since become more available, so we can now control what resources participants have access to. The Restricted track is closest to the original shared tasks, in that we specify precisely which annotated learner datasets participants should use, while the Unrestricted track allows use of any and all available datasets. The Low Resource track, in contrast, significantly limits the amount of annotated data available to participants and encourages development of systems that do not rely on large quantities of human-annotated sentences. A goal of the Low Resource track is thus to facilitate research into GEC for languages where annotated training corpora do not exist.

Like CoNLL-2014, the main evaluation metric was $F_{0.5}$, which weights precision twice as much as recall. Unlike CoNLL-2014 however, this

*Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

| | |
|---|---|
| Input | Travel by bus is exspensive , bored and annoying . |
| Output | Travelling by bus is expensive , boring and annoying . |

Table 1: An example input and output sentence.

is calculated using the ERRANT scorer (Bryant et al., 2017), rather than the $M^2$ scorer (Dahlmeier and Ng, 2012), because the ERRANT scorer can provide much more detailed feedback, e.g. in terms of performance on specific error types. Official evaluation is carried out on the Codalab competition platform, where a separate competition is created for each track. More details and links can be found on the official shared task website.[1]

The remainder of this report is structured as followed. Section 2 first summarises the task instructions and lists exactly what participants are asked to do. Section 3 next introduces the new W&I+LOCNESS corpus and describes how it was compiled. Section 3 also describes the other corpora that allowed in the shared task, including their formats and how they were standardised, and reports on a cross-corpora error type comparison for the first time. Section 4 next outlines each of the tracks and their restrictions, while Section 5 discusses the evaluation procedure. Section 6 next introduces the shared task participants and summarises each of their approaches, before Section 7 presents and analyses the final results. Appendix A contains more details about corpora and results.

## 2 Task Instructions

Participants are required to correct all grammatical, lexical and orthographic errors in written plain text files, one tokenised sentence per line, and are asked to produce equivalent corrected text files as output (Table 1). All text is tokenised using spaCy v1.9.0 and the `en_core_web_sm-1.2.0` model.[2]

Having produced a corrected text file, participants can then upload it to Codalab where it is automatically evaluated and a score returned. This procedure is the same for all tracks.

| | A | B | C | N | Total |
|---|---|---|---|---|---|
| **Train** | | | | | |
| Texts | 1,300 | 1,000 | 700 | - | 3,000 |
| Sentences | 10,493 | 13,032 | 10,783 | - | 34,308 |
| Tokens | 183,684 | 238,112 | 206,924 | - | 628,720 |
| **Dev** | | | | | |
| Texts | 130 | 100 | 70 | 50 | 350 |
| Sentences | 1,037 | 1,290 | 1,069 | 998 | 4,384 |
| Tokens | 18,691 | 23,725 | 21,440 | 23,117 | 86,973 |
| **Test** | | | | | |
| Texts | 130 | 100 | 70 | 50 | 350 |
| Sentences | 1,107 | 1,330 | 1,010 | 1,030 | 4,477 |
| Tokens | 18,905 | 23,667 | 19,953 | 23,143 | 85,668 |
| **Total** | | | | | |
| Texts | 1,560 | 1,200 | 840 | 100 | 3,700 |
| Sentences | 12,637 | 15,652 | 12,862 | 2,018 | 43,169 |
| Tokens | 221,280 | 285,504 | 248,317 | 46,260 | 801,361 |

Table 2: W&I (A, B, C) and LOCNESS (N) corpus statistics.

## 3 Data

This shared task introduces new annotated datasets: the Cambridge English Write & Improve (W&I) and LOCNESS corpus.

### 3.1 Cambridge English Write & Improve

Write & Improve[3] is an online web platform that assists non-native English students with their writing (Yannakoudakis et al., 2018). Specifically, students from around the world submit letters, stories, articles and essays in response to various prompts, and the W&I system provides automated feedback. Since 2014, W&I annotators have manually annotated some of these submissions with corrections and CEFR ability levels (Little, 2006).

#### 3.1.1 Corpus Compilation

Although users can submit any kind of text to the Write & Improve system, texts are first filtered before they are sent to the annotators to remove, for example, essay fragments, technical essays, copied website text, and non-English text. Although different versions of the same essays may be annotated to build up an annotated essay revision history, we only selected final revisions for inclusion in the W&I corpus.

We also ignored essays that met at least one of the following conditions:

- The text contained fewer than 33 words.
- More than 1.5% of all characters in the text were non-ASCII.
- More than 60% of all non-empty lines were both shorter than 150 characters and did not end with punctuation.

[1] https://www.cl.cam.ac.uk/research/nl/bea2019st/

[2] https://spacy.io/

[3] https://writeandimprove.com/

The precise values of these conditions were tuned manually such that they prioritised ‘cleaner’ texts while maintaining a large enough pool at each CEFR level. The last condition was designed to filter out texts that had been formatted to fit within a certain page width and so contained explicit new lines; e.g. “This is a \n broken sentence.” Such sentences were often tokenised incorrectly.

Since evaluation in GEC is typically carried out at the sentence level, we also wanted to make sure there was an even distribution of sentences at each CEFR level. We thus split the data on this basis, taking into account the fact that beginner essays tend to be shorter than more advanced essays. As CEFR levels are originally assigned at the essay level, sentence level CEFR labels are an approximation, and it is possible that the same sentence might receive a different label in a different text.

We ultimately selected 3,600 annotated submissions from W&I, which we distributed across training, development and test sets as shown in Table 2. We additionally annotated the test set a total of 5 times to better account for alternative corrections (cf. Bryant and Ng, 2015).

### 3.2 LOCNESS

Since most GEC research has traditionally focused on non-native errors, we also wanted to incorporate some native errors into the shared task. To do this, we used the LOCNESS corpus, a collection of approximately 400 essays written by native British and American undergraduates on various topics (Granger, 1998).[4]

Since these essays were typically much longer than the texts submitted to Write & Improve, we first filtered them to remove essays longer than 550 words. We also removed essays that contained transcription issue XML tags, such as `<quotation>` and `<illegible>`.

There are not enough essays to create an annotated LOCNESS training set, so we extracted a development and test set which was annotated by the W&I annotators. Like the W&I corpus, we also controlled the amount of native data in each set in terms of sentences to ensure a roughly even distribution at all levels. The test split was again annotated a total of 5 times to match the W&I test set. The statistics of this dataset are also shown in Table 2.

|  | Sentences | Tokens |
|---|---:|---:|
| FCE-train | 28,350 | 454,736 |
| FCE-dev | 2,191 | 34,748 |
| FCE-test | 2,695 | 41,932 |
| Lang-8 | 1,037,561 | 11,857,938 |
| NUCLE | 57,151 | 1,161,567 |

Table 3: FCE, Lang-8 and NUCLE corpus statistics.

### 3.3 Other Corpora

We allow the use of several existing learner corpora in the Restricted track of the shared task. Since these corpora were previously only available in different formats, we make new standardised versions available with the shared task (Table 3).

**FCE** The First Certificate in English (FCE) corpus is a subset of the Cambridge Learner Corpus (CLC) that contains 1,244 written answers to FCE exam questions (Yannakoudakis et al., 2011).

**Lang-8 Corpus of Learner English** Lang-8 is an online language learning website which encourages users to correct each other’s grammar. The Lang-8 Corpus of Learner English is a somewhat-clean, English subset of this website (Mizumoto et al., 2012; Tajiri et al., 2012). It is distinct from the raw, multilingual Lang-8 Learner Corpus.

**NUCLE** The National University of Singapore Corpus of Learner English (NUCLE) consists of 1,400 essays written by mainly Asian undergraduate students at the National University of Singapore (Dahlmeier et al., 2013). It is the official training corpus for the CoNLL-2013 and CoNLL-2014 shared tasks.

### 3.4 Corpus Standardisation

Since FCE and NUCLE were annotated according to different error type frameworks and Lang-8 and W&I+LOCNESS were not annotated with error types at all, we re-annotated all corpora automatically using ERRANT (Bryant et al., 2017). Specifically, we:

1. Tokenised the FCE and W&I+LOCNESS using spaCy v1.9.0. Lang-8 and NUCLE were pre-tokenised.

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

2. Used ERRANT to automatically classify the human edits in parallel FCE, NUCLE and W&I+LOCNESS sentences.
3. Used ERRANT to automatically extract and classify the edits in parallel Lang-8 sentences.

Note that as Lang-8 is not annotated with explicit edits, it only consists of parallel sentence pairs. We consequently used ERRANT to align the sentences and extract the edits automatically. While we could have also done the same for the other corpora, we instead chose to preserve and re-classify the existing human edits. Table 4 thus shows the ERRANT error type distributions for all these corpora, and makes them comparable for the first time.

In terms of edit operations, all corpora are fairly consistent with respect to the distribution of Missing (M) Replacement (R) and Unnecessary (U) word edits. Replacement edits are by far the most frequent category and account for roughly 60-65% of all edits in all datasets. Missing word edits account for roughly 20-25% of remaining edits, although this figure is noticeably lower in FCE and NUCLE. Unnecessary word edits account for 10-15% of all edits, although this figure rises to almost 20% in NUCLE. One possible explanation for this is that the NUCLE corpus also has more determiner (DET) errors, which are known to be problematic for Asian learners. Each corpus also contains roughly 2-3% of Unknown (UNK) edits that annotators identified but were unable to correct. UNK edits do not exist in Lang-8 because it was never annotated with edit spans.

NUCLE contains more than twice the proportion of noun number (NOUN:NUM) errors compared to the other corpora. This is possibly because noun number was one of the five error types targeted in the CoNLL-2013 shared task. Annotator focus might also account for the slightly higher proportion of determiner and subject-verb agreement (SVA) errors, which were also among the five targeted error types.

There is a significant difference in the proportion of punctuation (PUNCT) errors across corpora. Punctuation errors account for just 5% of all errors in NUCLE, but almost 20% in W&I. This is possibly because W&I contains data from a much wider range of learners than the other corpora. A similar pattern is observed with other (OTHER) errors, which account for over 25% of all errors in NUCLE and Lang-8, but roughly 13% of all errors in the FCE and W&I+LOCNESS. We surmise this is because edits are longer and noisier in the first two corpora (cf. Felice et al., 2016) and so do not fit into a more discriminative ERRANT error category.

```
S This are a sentence .
A 1 2|||R:VERB:SVA|||is|||-REQUIRED-|||NONE|||0
A 3 3|||M:ADJ|||good|||-REQUIRED-|||NONE|||0
A 1 2|||R:VERB:SVA|||is|||-REQUIRED-|||NONE|||1
A -1 -1|||noop|||-NONE-|||REQUIRED|||-NONE-|||2
```

Figure 1: Example M2 format with multiple annotators.

### 3.5 Data Formats

All the above corpora are released in M2 format, the standard format for annotated GEC files since the CoNLL-2013 shared task. The FCE and W&I+LOCNESS corpora are additionally released in an untokenised JSON format in case researchers want to inspect the raw data.

In M2 format (Figure 1), a line preceded by `S` denotes an original sentence while a line preceded by `A` indicates an edit annotation. Each edit line consists of the start and end token offsets of the edit, the error type, the tokenised correction string, a flag indicating whether the edit is required or optional, a comment field, and a unique annotator ID. The penultimate two fields are rarely used in practice however.

A 'noop' edit explicitly indicates when an annotator/system made no changes to the original sentence. If there is only one annotator, noop edits are optional, otherwise a noop edit should be included whenever at least 1 out of n annotators considered the original sentence to be correct. This is something to be aware of when combining individual M2 files, as missing noops can affect results.

Figure 1 can thus be interpreted as follows:

- Annotator 0 changed "are" to "is" and inserted "good" before "sentence" to produce the correction: "This is a good sentence ."
- Annotator 1 changed "are" to "is" to produce the correction: "This is a sentence ."
- Annotator 2 thought the original was correct and made no changes to the sentence: "This are a sentence ."

|  |  |  |  | W&I+LOCNESS |  |  |
|---|---|---|---|---|---|---|
|  | **FCE (all)** | **Lang-8** | **NUCLE** | **Train** | **Dev** | **Test** |
| **Type** | % | % | % | % | % | % |
| M | 21.00 | 26.41 | 19.09 | 25.29 | 26.32 | 24.86 |
| R | 64.39 | 59.99 | 59.04 | 61.43 | 61.23 | 63.40 |
| U | 11.47 | 13.60 | 19.31 | 10.69 | 10.21 | 10.34 |
| UNK | 3.13 | 0.00 | 2.57 | 2.59 | 2.24 | 1.41 |
| ADJ | 1.36 | 1.25 | 1.58 | 1.52 | 1.48 | 1.05 |
| ADJ:FORM | 0.28 | 0.19 | 0.27 | 0.24 | 0.21 | 0.18 |
| ADV | 1.94 | 3.37 | 1.95 | 1.51 | 1.51 | 1.45 |
| CONJ | 0.67 | 0.98 | 0.71 | 0.51 | 0.58 | 0.75 |
| CONTR | 0.32 | 0.99 | 0.11 | 0.30 | 0.39 | 0.32 |
| DET | 10.86 | 11.93 | 15.98 | 11.25 | 10.43 | 10.41 |
| MORPH | 1.90 | 1.62 | 3.14 | 1.85 | 2.07 | 2.50 |
| NOUN | 4.57 | 4.51 | 3.80 | 4.36 | 4.30 | 2.89 |
| NOUN:INFL | 0.50 | 0.18 | 0.12 | 0.12 | 0.13 | 0.28 |
| NOUN:NUM | 3.34 | 4.28 | 8.13 | 4.05 | 3.29 | 4.07 |
| NOUN:POSS | 0.51 | 0.35 | 0.61 | 0.60 | 0.87 | 0.93 |
| ORTH | 2.94 | 3.99 | 1.62 | 4.77 | 4.61 | 8.03 |
| OTHER | 13.26 | 26.62 | 25.65 | 12.76 | 12.84 | 15.69 |
| PART | 0.29 | 0.50 | 0.46 | 0.84 | 0.79 | 0.49 |
| PREP | 11.21 | 8.00 | 7.69 | 9.79 | 9.70 | 8.33 |
| PRON | 3.51 | 2.72 | 1.26 | 2.64 | 2.33 | 2.45 |
| PUNCT | 9.71 | 6.06 | 5.16 | 17.16 | 19.37 | 16.73 |
| SPELL | 9.59 | 4.45 | 0.26 | 3.74 | 5.07 | 4.63 |
| UNK | 3.13 | 0.00 | 2.57 | 2.59 | 2.24 | 1.41 |
| VERB | 7.01 | 6.52 | 4.31 | 5.86 | 5.27 | 5.09 |
| VERB:FORM | 3.55 | 2.56 | 3.49 | 3.56 | 3.09 | 3.10 |
| VERB:INFL | 0.19 | 0.15 | 0.01 | 0.04 | 0.07 | 0.12 |
| VERB:SVA | 1.52 | 1.58 | 3.47 | 2.23 | 1.94 | 2.28 |
| VERB:TENSE | 6.04 | 6.03 | 7.01 | 6.07 | 6.20 | 5.43 |
| WO | 1.82 | 1.18 | 0.66 | 1.64 | 1.25 | 1.40 |
| Total Edits | 52,671 | 1,400,902 | 44,482 | 63,683 | 7,632 | - |

Table 4: The ERRANT error type distributions of the FCE, Lang-8, NUCLE and W&I+LOCNESS corpora. See Bryant et al. (2017) for more information about each error type. The distribution of the W&I+LOCNESS test data is averaged across all 5 annotators.

## 4 Tracks

As parallel training data is now more readily available, a new feature of the BEA-2019 shared task is the introduction of three tracks: Restricted, Unrestricted and Low Resource. Each track controls the amount of *annotated* data that is available to participants. We place no restriction on the amount of *unannotated* data (e.g. for language modelling) or NLP tools (e.g. POS taggers, parsers, spellcheckers, etc.), provided the resources are publicly available.

### 4.1 Restricted Track

The Restricted Track is most similar to the previous shared tasks in that participants are limited to using only the official datasets as annotated training data (i.e. the FCE, Lang-8, NUCLE and W&I+LOCNESS). Since we do not limit unannotated data however, system submissions are still not entirely comparable given that they might use, for example, different amounts of monolingual or artificially-generated data.

### 4.2 Unrestricted Track

The Unrestricted Track is the same as the Restricted Track except participants may use any and all datasets and resources to build systems, including proprietary datasets and software. The main aim of this track is to determine how much better a system can do if it has access to potentially much larger amounts of data and/or resources.

### 4.3 Low Resource Track

The Low Resource Track is the same as the Restricted Track, except participants are only allowed to use the W&I+LOCNESS development set as annotated learner data. Since current GEC systems exploit as much annotated data as possible to reach the best performance, we hope this track will motivate work in GEC for other languages. We place no restriction on how participants use the W&I+LOCNESS development set; e.g. as a seed corpus to generate artificial data or to tune parameters to the shared task.

## 5 Evaluation

Systems are evaluated on the W&I+LOCNESS test set using the ERRANT scorer (Bryant et al., 2017), an improved version of the MaxMatch scorer (Dahlmeier and Ng, 2012) that was previously used in the CoNLL shared tasks. As in the previous shared tasks, this means that system performance is primarily measured in terms of span-based correction using the $F_{0.5}$ metric, which weights precision twice as much as recall.

In span-based correction, a system is only rewarded if a system edit exactly matches a reference edit in terms of both its token offsets and correction string. If more than one set of reference edits are available (there were 2 in CoNLL-2014 and 5 in BEA-2019), ERRANT chooses the reference that maximises the global $F_{0.5}$ score, or else maximises true positives and minimises false positives and false negatives. ERRANT is also able to report performance in terms of span-based detection and token-based detection (Table 5).

Although the W&I+LOCNESS training and development sets are released as separate files for each CEFR level, the test set texts are combined and shuffled such that the sentence order in each essay is preserved, but the order of the CEFR levels is random. This is done because systems should not expect to know the CEFR level of an input text in advance and should hence be prepared to handle all levels and abilities. In Section 7, we nevertheless also report system performance in terms of different CEFR and native levels, as well as in terms of detection and error types.

### 5.1 Metric Justification

Since robust evaluation is still a hot topic in GEC (cf. Asano et al., 2017; Choshen and Abend, 2018), we also wanted to provide some additional evidence that ERRANT $F_{0.5}$ is as reliable as MaxMatch $F_{0.5}$ and other popular metrics (Felice and Briscoe, 2015; Napoles et al., 2015). We evaluated ERRANT in relation to human judgements on the CoNLL-2014 test set using the same setup as Chollampatt and Ng (2018b), and found similar correlation coefficients (Table 6). Although this table shows that no metric is superior in all settings, the main advantage of ERRANT is that it can also provide much more detailed feedback than the alternatives; e.g. in terms of error types. We hope that researchers can make use of this information to build better systems.

## 6 Participants and Approaches

A total of 24 different teams took part in the BEA-2019 shared task across all 3 tracks. Of these, 21 submitted to the Restricted Track, 7 submitted to the Unrestricted Track, and 9 submitted to the Low Resource Track. This also meant 7 teams submitted to 2 separate tracks while 3 teams submitted to all 3 tracks.

Only 14 teams submitted system description papers however, with a further 4 sending short descriptions by email. The full list of teams, their approaches, and the data and resources they used in each track are shown in Table 8 (Appendix A.1). We refer the reader to the system description papers (where available) for more detailed information. Additionally: i) although Buffalo submitted to all 3 tracks, their paper does not describe their Low Resource system, ii) LAIX submitted exactly the same system to both the Restricted and Unrestricted Track, and iii) TMU submitted 2 separate papers about their respective Restricted and Low Resource Track systems.

While past GEC systems have employed different approaches, e.g. rules, classifiers, and statistical machine translation (SMT), in contrast, approximately two-thirds of all teams in the BEA-2019 shared task[5] used transformer-based neural machine translation (NMT) (Vaswani et al., 2017), while the remainder used convolutional neural networks (CNN), or both. Although they were most likely inspired by Junczys-Dowmunt et al. (2018) and Chollampatt and Ng (2018a), who previously reported state-of-the-art results on the CoNLL-2014 test set, the main consequence of this is that systems could only be differentiated based on lower-level system properties, such as:

- How much artificial data was used, if any, and how it was generated.
- How much over-sampled data was used, if any, and in what proportion.
- How many models were combined or ensembled.
- Whether system output was re-ranked.
- Whether the system contained an error detection component.

[5] Based on those that submitted system descriptions.

| **Original**<br>**Reference** | I often look at TV<br>[2, 4, watch] | **Span-based**<br>**Correction** | **Span-based**<br>**Detection** | **Token-based**<br>**Detection** |
|---|---|---|---|---|
| **Hypothesis 1** | [2, 4, watch] | Match | Match | Match |
| **Hypothesis 2** | [2, 4, see] | No match | Match | Match |
| **Hypothesis 3** | [2, 3, watch] | No match | No match | Match |

Table 5: Different types of evaluation in ERRANT.

| | **Corpus** | | **Sentence** |
|---|---|---|---|
| **Metric** | **Pearson r** | **Spearman** $\rho$ | **Kendall** $\tau$ |
| **ERRANT** | 0.64 | 0.626 | 0.623 |
| $\mathbf{M^2}$ | 0.623 | 0.687 | 0.617 |
| **GLEU** | 0.691 | 0.407 | 0.567 |
| **I-measure** | -0.25 | -0.385 | 0.564 |

Table 6: Correlation between various evaluation metrics and human judgements.

For example, Shuyao, UEDIN-MS and Kakao&Brain respectively trained their systems on 145 million, 100 million and 45 million artificial sentences, while CAMB-CUED instead concentrated on optimising the ratio of official to artificial sentences. TMU meanwhile focused entirely on re-ranking in their Restricted Track system, and AIP-Tohoku, CAMB-CLED, WebSpellChecker and YDGEC each incorporated sentence and/or token based detection components into their systems. Since most systems used different combinations of similar techniques, it is difficult to determine which were most successful. For example, several teams used artificial data, but they each generated it using different methods and corpora, so it is unclear which method performed best with respect to all the other uncontrolled system variables.

For the Low Resource track, many teams submitted the same Restricted Track systems except trained with the WikEd Corpus (Grundkiewicz and Junczys-Dowmunt, 2014) or other Wikipedia-based revision data. Notable exceptions include CAMB-CUED, who used Finite State Transducers (FST) to rank confusion sets with a language model; LAIX, who augmented their transformer NMT model with a series of 8 error-type specific classifiers; and TMU, who mapped 'cross-lingual' word embeddings to the same space to induce a phrase table for a SMT system. These systems hence represent promising alternatives in a heavily transformer NMT dominated shared task.

## 7 Results

All system output submitted to Codalab during the test phase was automatically annotated with ERRANT and compared against the gold standard references. Although this meant there was a mismatch between the automatically annotated hypotheses and the human annotated gold references, we deliberately chose this setting to remain faithful to the gold-standard training data and previous shared tasks. See Appendix A.7 for more on comparing gold and automatic references.

We also set a limit of a maximum of 2 submissions during the test phase to prevent teams from optimising on the test set. The best results, in terms of span-based correction ERRANT $F_{0.5}$, are used for the official BEA-2019 shared task results, and all scores are presented in Table 7.

### 7.1 Restricted Track - Overall

Since many teams used very similar approaches, it may be unsurprising that many of the Restricted Track scores were very similar. For example, the $F_{0.5}$ difference between the teams that ranked 3-5 was 0.17%, and the precision difference between the teams that ranked 4-6 was 0.47%. We thus carried out significance tests between all teams in each track using the bootstrap method (Efron and Tibshirani, 1993) based on $F_{0.5}$ (1,000 iterations, $p > .05$), and grouped systems that were not significantly different. The resulting groups showed that, for example, there was no significant difference between the top 2 teams and that the top 11 teams fit into 4 statistically significant groups. Groups were defined such that all teams in each group were statistically similar. This means, for example, that although ML@IITB was similar to YDGEC, it was different from Shuyao and the other teams in Group 2, and so was placed in Group 3 instead.

The top 2 teams in Group 1 scored significantly higher than all the teams in Group 2 most likely because both these teams 1) trained their systems on artificial data generated using error type distri-

**Restricted**

| Group | Rank | Teams | TP | FP | FN | P | R | $F_{0.5}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | UEDIN-MS | 3127 | 1199 | **2074** | 72.28 | 60.12 | **69.47** |
| | 2 | Kakao&Brain | 2709 | 894 | 2510 | **75.19** | 51.91 | 69.00 |
| 2 | 3 | LAIX | 2618 | 960 | 2671 | 73.17 | 49.50 | 66.78 |
| | 4 | CAMB-CLED | 2924 | 1224 | 2386 | 70.49 | 55.07 | 66.75 |
| | 5 | Shuyao | 2926 | 1244 | 2357 | 70.17 | 55.39 | 66.61 |
| | 6 | YDGEC | 2815 | 1205 | 2487 | 70.02 | 53.09 | 65.83 |
| 3 | 7 | ML@IITB | **3678** | 1920 | 2340 | 65.70 | **61.12** | 64.73 |
| | 8 | CAMB-CUED | 2929 | 1459 | 2502 | 66.75 | 53.93 | 63.72 |
| 4 | 9 | AIP-Tohoku | 1972 | 902 | 2705 | 68.62 | 42.16 | 60.97 |
| | 10 | UFAL | 1941 | 942 | 2867 | 67.33 | 40.37 | 59.39 |
| | 11 | CVTE-NLP | 1739 | 811 | 2744 | 68.20 | 38.79 | 59.22 |
| 5 | 12 | BLCU | 2554 | 1646 | 2432 | 60.81 | 51.22 | 58.62 |
| 6 | 13 | IBM | 1819 | 1044 | 3047 | 63.53 | 37.38 | 55.74 |
| 7 | 14 | TMU | 2720 | 2325 | 2546 | 53.91 | 51.65 | 53.45 |
| | 15 | qiuwenbo | 1428 | 854 | 2968 | 62.58 | 32.48 | 52.80 |
| 8 | 16 | NLG-NTU | 1833 | 1873 | 2939 | 49.46 | 38.41 | 46.77 |
| | 17 | CAI | 2002 | 2168 | 2759 | 48.01 | 42.05 | 46.69 |
| | 18 | PKU | 1401 | 1265 | 2955 | 52.55 | 32.16 | 46.64 |
| 9 | 19 | SolomonLab | 1760 | 2161 | 2678 | 44.89 | 39.66 | 43.73 |
| 10 | 20 | Buffalo | 604 | **350** | 3311 | 63.31 | 15.43 | 39.06 |
| 11 | 21 | Ramaiah | 829 | 7656 | 3516 | 9.77 | 19.08 | 10.83 |

**Unrestricted**

| Group | Rank | Teams | TP | FP | FN | P | R | $F_{0.5}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | LAIX | 2618 | 960 | 2671 | **73.17** | 49.50 | **66.78** |
| | 2 | AIP-Tohoku | 2589 | 1078 | 2484 | 70.60 | 51.03 | 65.57 |
| 2 | 3 | UFAL | 2812 | 1313 | 2469 | 68.17 | 53.25 | 64.55 |
| 3 | 4 | BLCU | **3051** | 2007 | **2357** | 60.32 | **56.42** | 59.50 |
| 4 | 5 | Aparecium | 1585 | 1077 | 2787 | 59.54 | 36.25 | 52.76 |
| 5 | 6 | Buffalo | 699 | **374** | 3265 | 65.14 | 17.63 | 42.33 |
| 6 | 7 | Ramaiah | 1161 | 8062 | 3480 | 12.59 | 25.02 | 13.98 |

**Low Resource**

| Group | Rank | Teams | TP | FP | FN | P | R | $F_{0.5}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | UEDIN-MS | 2312 | 982 | **2506** | **70.19** | **47.99** | **64.24** |
| 2 | 2 | Kakao&Brain | **2412** | 1413 | 2797 | 63.06 | 46.30 | 58.80 |
| 3 | 3 | LAIX | 1443 | **884** | 3175 | 62.01 | 31.25 | 51.81 |
| | 4 | CAMB-CUED | 1814 | 1450 | 2956 | 55.58 | 38.03 | 50.88 |
| 4 | 5 | UFAL | 1245 | 1222 | 2993 | 50.47 | 29.38 | 44.13 |
| 5 | 6 | Siteimprove | 1299 | 1619 | 3199 | 44.52 | 28.88 | 40.17 |
| | 7 | WebSpellChecker | 2363 | 3719 | 3031 | 38.85 | 43.81 | 39.75 |
| 6 | 8 | TMU | 1638 | 4314 | 3486 | 27.52 | 31.97 | 28.31 |
| 7 | 9 | Buffalo | 446 | 1243 | 3556 | 26.41 | 11.14 | 20.73 |

Table 7: Official BEA-2019 results for all teams in all tracks using the main overall span-based correction ERRANT $F_{0.5}$. The highest values (lowest for False Positives and False Negatives) are shown in bold.

butions and confusion sets, and 2) re-ranked their system output. In contrast, Shuyao used a similar method to generate artificial data, but did not rerank, while CAMB-CLED used back-translation to generate artificial data, but did re-rank. This suggests that confusion set approaches to artificial data generation are more successful than back-translated approaches.

### 7.2 Unrestricted Track - Overall

Since participants were allowed to use any and all datasets in the Unrestricted Track, we expected scores to be higher, but the highest scoring team actually submitted exactly the same system to the Unrestricted Track as they did to the Restricted Track. The top 2 teams in the Restricted Track could thus also have scored highest on this track if they did the same.

Of the remaining teams, AIP-Tohoku and UFAL increased their scores by approximately 5 $F_{0.5}$ using non-public Lang-8 and parallel Wikipedia data respectively, BLCU added a more modest 1 $F_{0.5}$ similarly using non-public Lang-8 data, and Buffalo added roughly 3 $F_{0.5}$ using artificial data generated from a subsection of the English Gigaword corpus (Graff and Cieri, 2003). While it is unsurprising that larger quantities of training data tended to lead to higher scores, these results help quantify the extent to which performance can be improved by simply adding more data.

### 7.3 Low Resource Track - Overall

The teams that came top of the Restricted Track also dominated in the Low Resource Track. The UEDIN-MS system even outperformed 14 of the Restricted Track submissions despite the limited training data. This is most likely because UEDIN-MS and Kakao&Brain both made effective use of artificial data.

The CAMB-CUED system also achieved a fairly competitive score despite not using any parallel training data. This contrasts with LAIX, who scored higher by 1 $F_{0.5}$ using a complicated system of classifiers, CNNs and transformer NMT models. The TMU system is also notable for applying techniques from unsupervised SMT to GEC for the first time (cf. Artetxe et al., 2018). Although it performed poorly overall, it took several years to adapt supervised SMT to GEC (Junczys-Dowmunt and Grundkiewicz, 2016), so we hope researchers will continue to explore unsupervised SMT in future work.

## 8 Conclusion

It is undeniable that significant progress has been made since the last shared task on grammatical error correction five years ago. Transformer based neural machine translation proved effective, and teams generally scored significantly higher in BEA-2019 than in the previous CoNLL-2014 shared task. This is significant because we also introduced a new corpus, the Cambridge English Write & Improve + LOCNESS corpus, which contains a much wider range of texts at different ability levels than previous corpora, yet systems still generalised well to this much more diverse dataset.

Overall, the most successful systems were submitted by UEDIN-MS (Grundkiewicz et al., 2019) and Kakao&Brain (Choe et al., 2019) who respectively ranked first and second in both the Restricted and Low Resource Tracks. UEDIN-MS additionally scored just 5 $F_{0.5}$ lower in the Low Resource Track (64.24) than the Restricted Track (69.47), which shows that it is possible to build a competitive GEC system without large quantities of human annotated training data.

Finally, we note that the appendix contains a much more fine-grained analysis of system performance in terms of CEFR levels, edit operations, error types, single vs. multi token errors, detection vs. correction, and a comparison with other metrics. Some key findings include:

- There was a clear indication that different systems performed better at different CEFR levels.
- All systems still struggle most with content word errors.
- Systems are significantly better at correcting multi token errors than they were 5 years ago.
- The GLEU metric (Napoles et al., 2015) strongly correlates with recall and seems to be less discriminative than other metrics.

We ultimately hope that the results and corpus statistics we report will serve as useful benchmarks and guidance for future work.

## Acknowledgements

We are extremely grateful to Diane Nicholls and her team of annotators for annotating the entire W&I+LOCNESS corpus. Our work was supported by Cambridge Assessment, University of Cambridge.

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Hiroki Asano, Tomoya Mizumoto, and Kentaro Inui. 2017. Reference-based metrics can be replaced with reference-less metrics in evaluating grammatical error correction systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–348. Asian Federation of Natural Language Processing.

Hiroki Asano, Tomoya Mizumoto, and Masato Mita. 2019. The AIP-Tohoku System at the BEA-2019 Shared Task. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. 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.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 697–707, Beijing, China. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A Neural Grammatical Error Correction System Built On Better Pre-training and Sequential Transfer Learning. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018a. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Shamil Chollampatt and Hwee Tou Ng. 2018b. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741. Association for Computational Linguistics.

Leshem Choshen and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics.

Bohdan Didenko and Julia Shaptala. 2019. Multiheaded Architecture Based on BERT for Grammar Error Correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

B. Efron and R.J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587. Association for Computational Linguistics.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.

David Graff and Christopher Cieri. 2003. English gigaword ldc2003t05.

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. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *Advances in Natural Language Processing*, pages 478–490, Cham. Springer International Publishing.

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.

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 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. 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.

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 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. Learning to combine Grammatical Error Corrections. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2019. (Almost) Unsupervised Grammatical Error Correction using a Synthetic Comparable Corpus. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Ophélie Lacroix, Simon Flachs, and Anders Søgaard. 2019. Noisy Channel for Low Resource Grammatical Error Correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Ruobing Li, Chuan Wang, Yefei Zha, Yonghong Yu, Shiman Guo, Qiang Wang, Yang Liu, and Hui Lin. 2019. The LAIX Systems in the BEA-2019 GEC Shared Task. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar. 2018. Weakly supervised grammatical error correction using iterative decoding. *CoRR*, abs/1811.01710.

David Little. 2006. The common european framework of reference for languages: Content, purpose, origin, reception and impact. *Language Teaching*, 39(3):167190.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012: Posters*, pages 863–872. The COLING 2012 Organizing Committee.

Jakub Náplava and Milan Straka. 2019. CUNI System for the Building Educational Applications 2019 Shared Task: Grammatical Error Correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593. Association for Computational Linguistics.

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. 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. Association for Computational Linguistics.

Mengyang Qiu, Xuejiao Chen, Maggie Liu, Krishna Parvathala, Apurva Patil, and Jungyeul Park. 2019. Improving Precision of Grammatical Error Correction with Cheat Sheet. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

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.

Felix Stahlberg and Bill Byrne. 2019. The CUED's Grammatical Error Correction Systems for BEA-2019. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Singsound System for GEC-2019. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Liner Yang and Chencheng Wang. 2019. The BLCU System in the BEA 2019 Shared Task. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

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, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. 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. *CoRR*, abs/1903.00138.

# A Appendix

## A.1 Participants and Approaches

| Team | Track | Approach | Data | Resources |
|---|---|---|---|---|
| AIP-Tohoku<br>(Asano et al., 2019) | R | Multi-task learning was used to predict the CEFR level of a sentence and whether it contained any errors. Flagged sentences were then corrected by a transformer NMT model trained on official and back-translated data. | Artificial data, ASAP, ICLE, ICNALE, TOEFL11, Simple Wiki$_{en}$ | BERT, Fairseq |
| | U | The same as above, except the official data was augmented with EFCAMDAT and Lang-8$_{priv}$. | EFCAMDAT, Lang-8$_{priv}$ | |
| Aparecium | U | - | - | - |
| BLCU<br>(Yang and Wang, 2019) | R | Spelling errors were corrected by a context sensitive spellchecker and the output was pipelined to an ensemble of eight transformer NMT systems. The NMT systems were trained on artificial data and fine-tuned on official data. The artificial data was generated by uniformly inserting/replacing/deleting 30% of all tokens in monolingual sentences. | Artificial data, 1BW | CyHunspell, Fairseq, KenLM |
| | U | The same as above, except the official data was augmented with ∼6 million non-public Lang-8 sentences. | Lang-8$_{priv}$ | |
| Buffalo<br>(Qiu et al., 2019) | R | An SMT and CNN encoder-decoder system were respectively trained on official data that had been augmented with repetitions of phrase pair edits in context. | - | GIZA++, KenLM, Moses |
| | U | The same as above, except the system was additionally trained on artificial data that targeted six error types. | Artificial data, AFP News (English Gigaword) | |
| | LR | - | - | - |
| CAI | R | - | - | - |
| CAMB-CLED<br>(Yuan et al., 2019) | R | A CNN encoder-decoder with auxiliary token-based and sentence-based detection functions was combined with an ensemble of four transformer NMT and language models trained on official and back-translated data. The n-best lists from each system were combined using an FST and then re-ranked. | Artificial data, 1BW, BNC, ukWaC, Wiki$_{en}$ | ELMo, fastText, GloVe, Hunspell, Jamspell, KenLM, OpenFST, SGNMT, Stanford Parser, Tensor2Tensor |
| CAMB-CUED<br>(Stahlberg and Byrne, 2019) | R | An ensemble of four transformer NMT models were trained on different combinations of over-sampled official data and back-translated artificial data. The best result came from a ratio of approx. 1:8 official to artificial sentences. | Artificial data, English News Crawl | SGNMT, Tensor2Tensor |
| | LR | Confusion sets were generated using CyHunspell, a database of morphological inflections, and manual rules. The confusion sets were combined in a cascade of FSTs that were weighted by edit type. The resulting FST was finally constrained by a transformer language model. | English News Crawl | AGID, CyHunspell, OpenFST, SGNMT, Tensor2Tensor |
| CVTE-NLP<br>Email | R | A CNN encoder-decoder based on Chollampatt and Ng (2018a). | - | - |
| IBM<br>(Kantor et al., 2019) | R | A novel spellchecker, a language model based confusion set replacement system, and four transformer NMT systems trained on over-sampled official and artificial data were all combined using a system combination technique. The new spellchecker performed better than other popular alternatives. The artificial data was generated based on the error distribution in the W&I+LOCNESS development set. | Project Gutenberg, | BERT, LibreOffice dictionaries, Nematus |

| | | | | |
|---|---|---|---|---|
| Kakao&Brain<br>(Choe et al., 2019) | R | Spelling and orthography errors were corrected by a context sensitive spellchecker and the output was pipelined to an ensemble of five transformer NMT systems trained on official and artificial data. The artificial data was generated by applying edits that occurred more than four times in the W&I+LOCNESS dev set, or else occurred in a POS-based confusion set, to several native corpora (45 million sentences) based on the edit probability. The output was then re-ranked and type-filtered. | Artificial data, Gutenberg, Tatoeba, WikiText-103 | Fairseq, Hunspell, SentencePiece |
| | LR | The same as above, except the artificial data was only generated from the W&I+LOCNESS dev set and the system was not fine-tuned. | | |
| LAIX<br>(Li et al., 2019) | R, U | An ensemble of four CNN-based systems and an ensemble of eight transformer NMT models were combined using different system combination techniques. | Common Crawl | - |
| | LR | Eight bi-directional GRU classifiers were trained to correct eight different error types. An ensemble of CNN and transformer NMT models were also trained on the WikEd corpus. The systems were combined and augmented with some rules and a spellchecker. | Common Crawl, WikEd, Wiki$_{en}$ | Enchant |
| ML@IITB<br>Email | R | A sequence labelling model proposed edits in-place rather than regenerating the whole sentence. In-place edits were restricted to the top 500 most common insertions and replacements in the training data, as well as a set of morphological transformations and spelling errors. | - | autocorrect, BERT |
| NLG-NTU<br>Email | R | A CNN encoder-decoder based on Wu et al. (2019). | - | Fairseq |
| PKU | R | - | - | - |
| qiuwenbo | R | - | - | - |
| Ramaiah | R, U | - | - | - |
| Shuyao<br>(Xu et al., 2019) | R | A comprehensive collection of confusion sets were designed and used to generate artificial data based on various properties of the official data. An ensemble of four transformer NMT models were trained on the artificial data and fine-tuned on the official data. | Artificial data, 1BW, English News Crawl | NLTK, Tensor2Tensor |
| Siteimprove<br>(Lacroix et al., 2019) | LR | Confusion sets were generated from the WikEd corpus and filtered to remove noise. Each candidate in each confusion set was then evaluated by two pretrained language models (BERT and GPT-2) in a noisy channel model. Beam search was used to account for interacting errors. | WikEd, Wiki$_{en}$ | BERT, Enchant, GPT-2, Spacy, Unimorph, Wiktionary, Wordnet |
| SolomonLab | R | - | - | - |
| TMU<br>(Kaneko et al., 2019; Katsumata and Komachi, 2019) | R | An ensemble of three transformer NMT models were trained on the official data, and the output was re-ranked using the same features as Chollampatt and Ng (2018a) along with a new BERT feature. | Common Crawl, Wiki$_{en}$ | BERT, Fairseq, fastText, pyspellchecker |
| | LR | Finnish News Crawl was translated to English using Google Translate (source data). Word embeddings were spearately trained on this and native English News Crawl (target data). The source and target word embedding were then mapped to the same 'cross-lingual' space, which was used to induce a phrase table for a SMT system. | 1BW, English News Crawl, Finnish News Crawl | FastAlign, Google Translate, KenLM, Moses, pyspellchecker |
| UEDIN-MS<br>(Grundkiewicz et al., 2019) | R | Artificial errors generated from Aspell confusion sets were introduced to 100 million English News Crawl sentences based on the error distribution of the W&I+LOCNESS development set. An ensemble of eight transformer NMT systems were trained on over-sampled official and artificial data and the output was re-ranked. | Artificial data, English News Crawl | Aspell, Enchant, KenLM, Marian, SentencePiece |
| | LR | Similar to the above, except the official data was substituted with a filtered version of the WikEd corpus. | WikEd | |

| | | | | |
|---|---|---|---|---|
| UFAL (Náplava and Straka, 2019) | R | A transfomer NMT model was trained on over-sampled official data and fine-tuned with dropout, checkpoint averaging and iterative decoding. | - | Tensor2Tensor |
| | U | The same as the low resource track system, except fine-tuned on the over-sampled official data. | Wiki$_{en}$ | |
| | LR | The same as the restricted track system, except the official data was substituted for consecutive English Wikipedia snapshots with added character perturbation. | | |
| WebSpellChecker (Didenko and Shaptala, 2019) | LR | A transformer model is used to detect errors based on edit operation, error type, or predicted correction method. The model predicts a start and end offset for an edit, and a correction model predicts the most likely correction for this span from a LM vocabulary. | - | BERT |
| YDGEC Email | R | A pipeline of: 1. A spelling correction model, 2. A sentence-level detection model to filter out correct sentences, 3. Three error type models for each of missing articles, punctuation and SVA, 4. An ensemble of four transformer NMT models trained on different combinations of over-sampled training data. 5. Re-ranking. | Wiki$_{en}$ | BERT, Marian |

Table 8: This table shows all the teams that participated in the BEA-2019 shared task and attempts to summarise their approaches in each track. R, U ad LR respectively denote the Restricted, Unrestricted and Low Resource tracks. All teams used the permitted official datasets in all their submissions, so this information is not included in the Data column. See each system paper (where available) for more information about each of the other datasets and/or resources used by each team. A dash (-) indicates either that there is no information for the given cell, or else no additional datasets or resources were used.

### A.2 CEFR Levels

Since one of the main contributions of the BEA-2019 shared task was the introduction of new data annotated for different proficiency levels, we analysed each team in terms of their CEFR and Native level performance. The $F_{0.5}$ results for each team and level are thus plotted in Figure 2.

The top 10 teams in the Restricted Track all performed best on C level texts, while the bottom 11 systems typically performed best on A level texts: a clear indication that some systems are more biased towards different learner levels than others. Different systems may also be differently suited to correcting different error types. For example, while punctuation errors are fairly rare at levels A and B, they are much more common at levels C and N. Conversely, noun number errors are common at levels A and B, but are rarer at levels C and N. Consequently, system performance at different CEFR levels is affected by each system's ability to correct specific error types.

The bottom 13 teams in the Restricted Track also typically struggled most with the native level texts. For example, there is an almost 15 $F_{0.5}$ gap between AIP-Tohoku's N level result and their next lowest CEFR level. Since we did not release any native level training data, we note that some systems failed to generalise to the levels and domains that they could not train on. In contrast, Low Resource Track submissions tended to score highest on native level texts, perhaps because several were trained on corrupted native data which may be more similar to the N level texts than the genuine learner data.

### A.3 Edit Operation

Results for each team in terms of Missing, Replacement and Unnecessary word errors are shown in Table 9. These results mainly provide a high level overview of the types of errors systems were able to correct, but can also be used to help identify different system strengths and weaknesses. For example, UEDIN-MS only ranked 7th in terms of correcting missing word errors, but made up for this by scoring much higher at replacement and unnecessary word errors, suggesting their system could be improved by paying more attention to missing word errors.

In contrast, Kakao&Brain scored highest at missing word errors, but came 2nd in terms of replacement word errors and 7th in terms of unnecessary word errors. Although they also achieved the highest precision out of all teams in terms of unnecessary word errors, they did so at the cost of almost half the recall of the UEDIN-MS system. This suggest that Kakao&Brain should instead focus on improving unnecessary word error correction. That said, it is also worth reiterating that approximately 65% of all errors are replacement word errors, compared to 25% missing and 10% unnecessary, and so it is arguably more important to focus on replacement word errors more than any other category.

In the Restricted Track, ML@IITB and BLCU respectively scored highest in terms of recall on missing and unnecessary word errors. This perhaps suggests that ML@IITB's strategy of only paying attention to the top 500 most frequent missing word errors paid off, while BLCU's artificial data generation method treated all edit operations equally, and so was perhaps more highly optimised for unnecessary word errors.

In the Low Resource Track, UEDIN-MS was again the dominant system in terms of replacement and unnecessary word errors, although Kakao&Brain again came top in terms of missing word errors. There was also a larger discrepancy between certain teams' operation scores and, for example, UFAL scored 43.36 and 50.91 $F_{0.5}$ on missing and replacement word errors, but just 14.89 $F_{0.5}$ on unnecessary word errors, while WebSpellChecker scored 60.40 $F_{0.5}$ on missing word errors, but just 34.13 and 28.63 on replacement and unnecessary word errors. These results suggest that some systems are more heavily biased towards some edit operations than others, but researchers can hopefully use this information to overcome their system's weaknesses.

### A.4 Single vs. Multi Token Edits

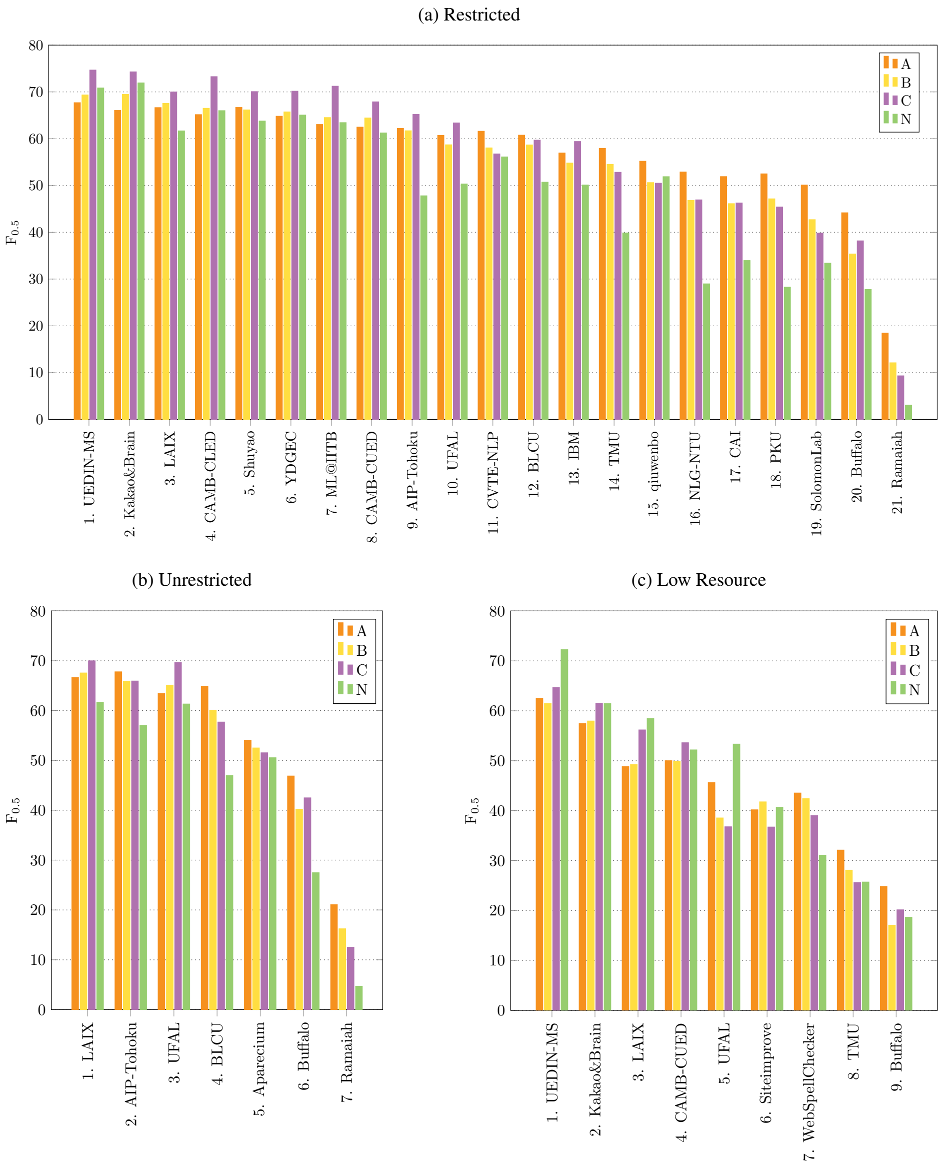
In addition to error types, we also examined system performance in terms of single and multi token edits, where a multi token edit is defined as any edit that contains 2 or more tokens on at least one side of the edit; e.g. [*eat* → *has eaten*] or [*only can* → *can only*]. Systems were evaluated in this setting mainly because Sakaguchi et al. (2016) previously advocated fluent, rather than simply grammatical, edits in GEC, yet fluency edits often involve multi token corrections. When Bryant et al. (2017) evaluated the CoNLL-2014 systems in terms of multi token edits however, they found

| **Restricted** | **M** | | | **R** | | | **U** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Team** | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ |
| UEDIN-MS | 70.20 | 64.38 | 68.95 | **73.10** | **58.42** | **69.60** | 73.10 | 60.23 | **70.11** |
| Kakao&Brain | **79.39** | 65.70 | **76.22** | 72.51 | 47.83 | 65.73 | **76.33** | 33.91 | 61.05 |
| LAIX | 79.01 | 58.79 | 73.93 | 70.22 | 46.67 | 63.78 | 73.68 | 40.66 | 63.39 |
| CAMB-CLED | 73.30 | 64.32 | 71.31 | 69.88 | 50.27 | 64.82 | 66.05 | 59.51 | 64.63 |
| Shuyao | 75.53 | 61.14 | 72.14 | 67.36 | 53.13 | 63.94 | 72.61 | 53.62 | 67.81 |
| YDGEC | 75.72 | 59.99 | 71.95 | 69.60 | 47.86 | 63.80 | 60.92 | 64.17 | 61.54 |
| ML@IITB | 74.05 | **73.37** | 73.91 | 63.87 | 53.36 | 61.45 | 53.78 | 67.98 | 56.13 |
| CAMB-CUED | 67.81 | 66.84 | 67.62 | 66.35 | 47.59 | 61.50 | 65.40 | 56.47 | 63.39 |
| AIP-Tohoku | 71.56 | 48.63 | 65.39 | 69.26 | 37.73 | 59.34 | 61.56 | 54.70 | 60.05 |
| UFAL | 71.02 | 47.76 | 64.72 | 66.11 | 36.73 | 56.99 | 64.72 | 45.31 | 59.61 |
| CVTE-NLP | 68.50 | 40.22 | 60.05 | 68.96 | 38.55 | 59.56 | 62.91 | 37.20 | 55.27 |
| BLCU | 63.86 | 50.21 | 60.57 | 63.16 | 48.36 | 59.52 | 50.48 | **68.02** | 53.23 |
| IBM | 71.88 | 48.40 | 65.52 | 59.56 | 33.58 | 51.58 | 61.70 | 31.59 | 51.82 |
| TMU | 63.85 | 57.26 | 62.42 | 52.55 | 49.32 | 51.87 | 42.79 | 52.94 | 44.50 |
| qiuwenbo | 58.94 | 25.99 | 47.01 | 64.64 | 34.34 | 54.95 | 56.04 | 33.63 | 49.45 |
| NLG-NTU | 56.68 | 41.46 | 52.80 | 48.74 | 36.09 | 45.55 | 41.80 | 45.66 | 42.52 |
| CAI | 55.59 | 48.01 | 53.89 | 46.81 | 39.45 | 45.12 | 39.64 | 44.04 | 40.45 |
| PKU | 66.60 | 35.43 | 56.64 | 49.39 | 30.16 | 43.80 | 48.15 | 38.41 | 45.82 |
| SolomonLab | 53.18 | 25.38 | 43.62 | 45.62 | 44.18 | 45.33 | 33.72 | 38.26 | 34.54 |
| Buffalo | 57.43 | 7.38 | 24.37 | 64.24 | 17.62 | 42.01 | 62.24 | 16.22 | 39.71 |
| Ramaiah | 47.31 | 28.04 | 41.59 | 6.23 | 14.71 | 7.04 | 11.69 | 27.50 | 13.21 |

| **Unrestricted** | **M** | | | **R** | | | **U** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Team** | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ |
| LAIX | **79.01** | **58.79** | **73.93** | 70.22 | 46.67 | 63.78 | **73.68** | 40.66 | **63.39** |
| AIP-Tohoku | 72.23 | 54.83 | 67.92 | **72.70** | 47.38 | **65.68** | 60.47 | 63.59 | 61.07 |
| UFAL | 69.21 | 54.28 | 65.60 | 69.47 | 51.38 | 64.90 | 61.03 | 61.23 | 61.07 |
| BLCU | 64.61 | 53.85 | 62.13 | 63.27 | **54.50** | 61.30 | 47.26 | **70.93** | 50.64 |
| Aparecium | 63.61 | 38.29 | 56.18 | 58.89 | 37.36 | 52.80 | 53.33 | 24.02 | 42.87 |
| Buffalo | 70.64 | 9.49 | 30.87 | 65.03 | 19.65 | 44.49 | 61.24 | 20.26 | 43.60 |
| Ramaiah | 55.06 | 31.99 | 48.12 | 9.14 | 22.02 | 10.35 | 11.53 | 28.48 | 13.09 |

| **Low Resource** | **M** | | | **R** | | | **U** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Team** | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ | **P** | **R** | **F**$_{0.5}$ |
| UEDIN-MS | 69.65 | 55.92 | 66.39 | **71.56** | **46.77** | **64.70** | **61.16** | 33.11 | **52.30** |
| Kakao&Brain | 70.12 | **61.76** | **68.27** | 59.00 | 41.10 | 54.28 | 60.98 | 31.45 | 51.33 |
| LAIX | 68.19 | 41.30 | 60.33 | 59.11 | 27.03 | 47.77 | 59.07 | 31.32 | 50.18 |
| CAMB-CUED | 55.05 | 22.13 | 42.42 | 57.65 | 41.97 | 53.64 | 46.46 | 45.30 | 46.22 |
| UFAL | 57.82 | 21.68 | 43.36 | 58.43 | 33.61 | 50.91 | 14.64 | 16.01 | 14.89 |
| Siteimprove | **80.10** | 17.16 | 46.21 | 42.76 | 33.33 | 40.47 | 34.78 | 22.71 | 31.44 |
| WebSpellChecker | 60.72 | 59.14 | 60.40 | 33.96 | 34.80 | 34.13 | 25.65 | **53.63** | 28.63 |
| TMU | 35.25 | 58.81 | 38.32 | 21.78 | 18.53 | 21.04 | 18.74 | 26.72 | 19.93 |
| Buffalo | 27.22 | 11.94 | 21.67 | 26.22 | 11.95 | 21.16 | 25.00 | 3.05 | 10.24 |

Table 9: This table shows the performance of each team in each track in terms of Missing, Replacement and Unnecessary token edits. In terms of frequency, approximately 25% of all edits are M, 65% are R, and 10% are U (cf. Table 4). The highest scores for each column are shown in bold.


(a) Restricted

(b) Unrestricted

(c) Low Resource

Figure 2: The $F_{0.5}$ scores for each team in each track in terms of CEFR and native levels: A (beginner), B (intermediate), C (advanced) and N (native).

that only 3 out of 12 teams achieved scores higher than 10 $F_{0.5}$, prompting them to conclude that significant progress must be made before fluency corrections become a viable option.

With this in mind, we are pleased to report that Figure 3 shows systems have indeed made significant progress in terms of correcting multi token edits, and in fact almost all teams scored higher than 20 $F_{0.5}$, with an average of 42 $F_{0.5}$. While systems still scored higher in terms of single token errors overall, this is most likely because single token errors are not only typically easier to correct than multi token errors, but are also much more frequent and tend to account for roughly 70-80% of all edits.

It is also noteworthy that Kakao&Brain actually surpassed UEDIN-MS in terms of single token error performance in the Restricted Track, but fell much shorter in terms of multi token edits. Shuyao was also particularly adept at correcting multi token errors, coming second after UEDIN-MS overall. In the Low Resource track meanwhile, Siteimprove is notable for not correcting any multi token errors at all, however this was because their system only targeted a limited number of single token error types by design.

## A.5 Detection vs. Correction

One aspect of system performance that is seldom reported in the literature is that of error detection;

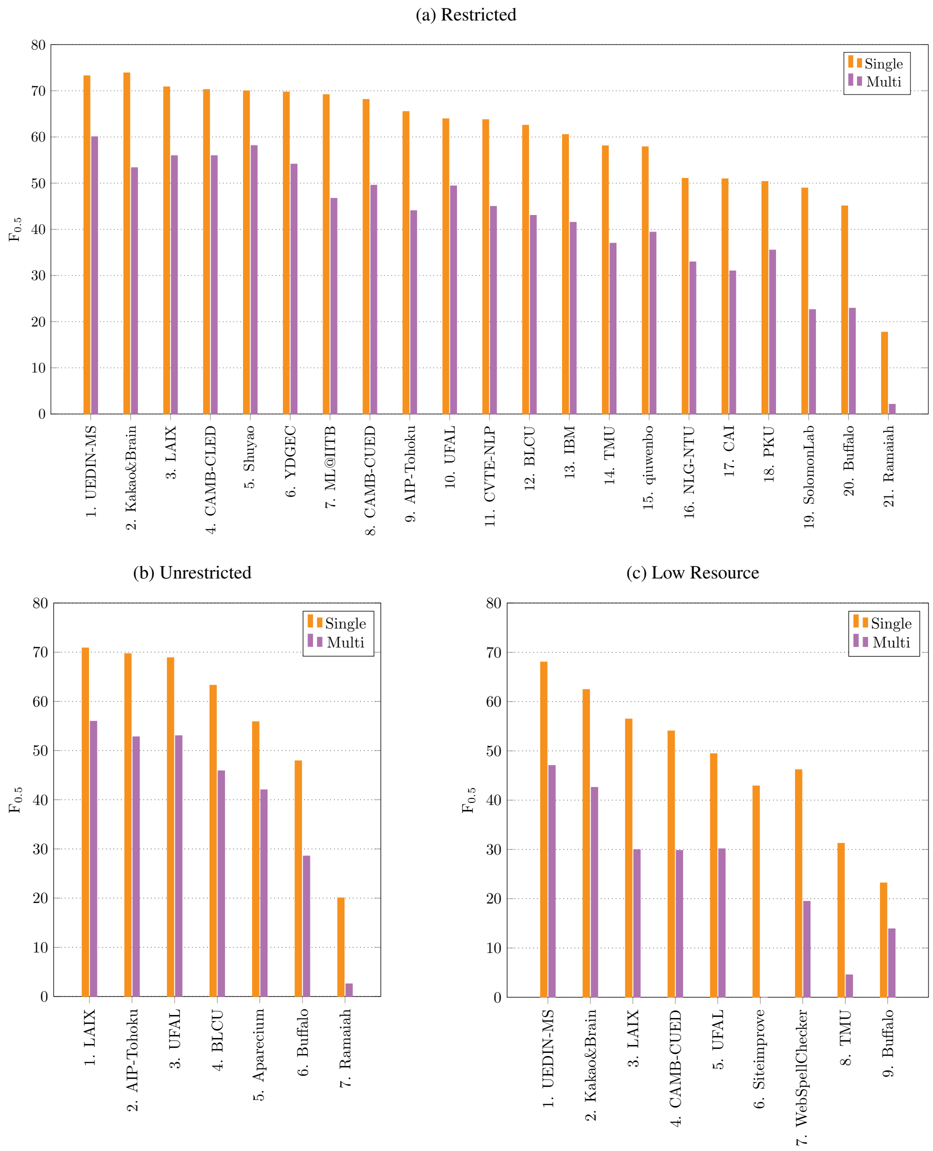


Figure 3: The $F_{0.5}$ scores for each team in each track in terms of single and multi token edits. A multi token is defined as any edit that has 2 or more tokens on at least one side of the edit.

i.e. the extent to which a system can identify errors. This is significant because detection is an important task in its own right as well as the first step in GEC. Figure 4 compares each team in terms of span based detection, span based detection and token based correction $F_{0.5}$.

In general, all systems were fairly consistent in terms of the difference between their detection and correction scores, with most teams scoring approximately 12-17 $F_{0.5}$ higher on token based detection than correction. CAMB-CLED and ML@IITB are noteworthy for achieving the 2nd and 3rd highest scores in terms of token detection, although the former can be explained by the fact that CAMB-CLED explicitly modelled detection in their approach. One of the main applications of this graph is thus to inform teams whether they should focus on improving the correction of errors they can already detect, or else extend their systems to detect new errors.

## A.6 Main Error Types

The overall $F_{0.5}$ scores for each of the main 24 ERRANT error types for each team in the Restricted Track are shown in Table 10, while similar results for the Unrestricted and Low Resource Tracks are shown in Table 11. The cells in these tables have also been shaded such that a darker red indicates a lower score. This makes it easier to see at a glance which error types were the hardest for all systems to correct.

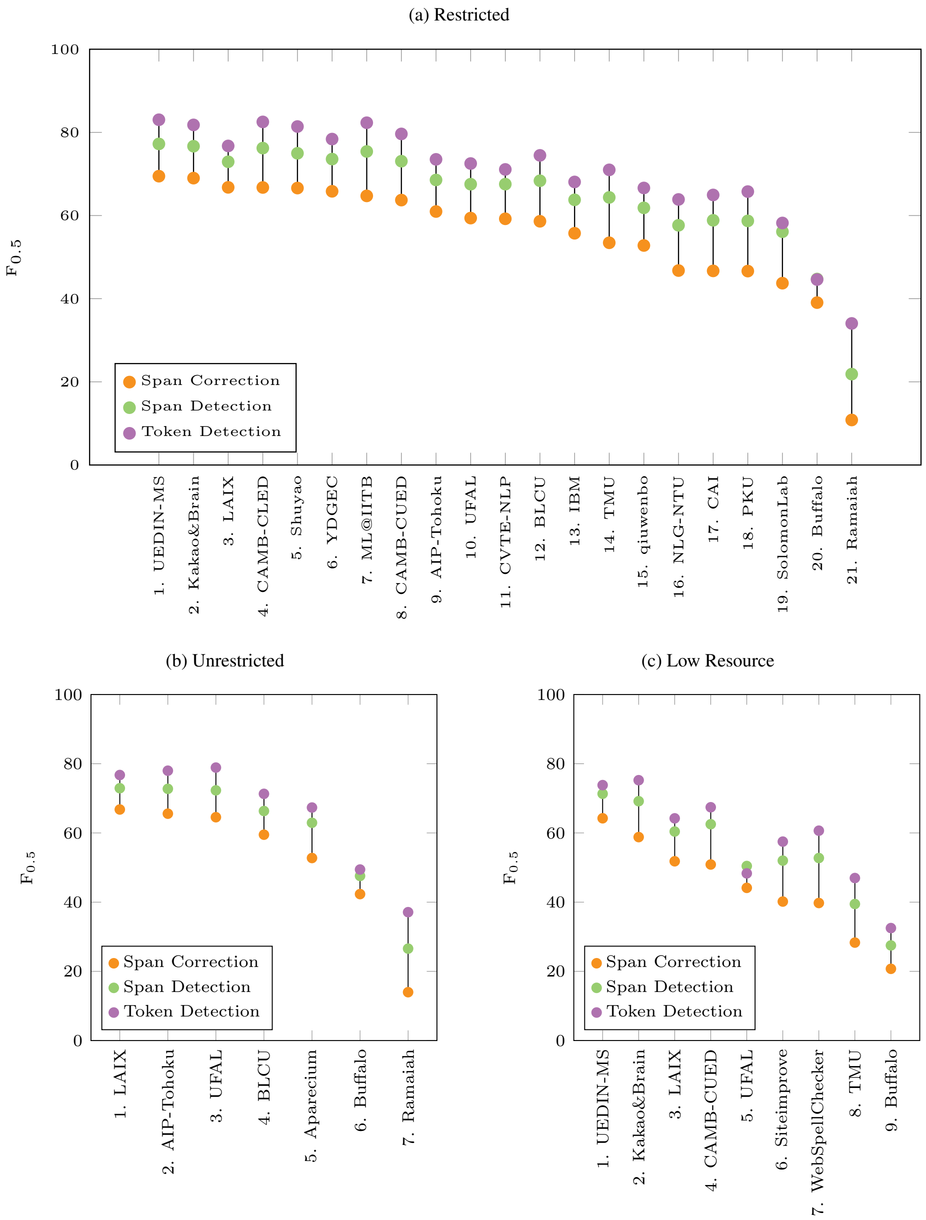


Figure 4: The difference in $F_{0.5}$ scores in terms of span based correction, span based detection, and token based detection (as defined in Section 5) for each team in each track.

With this in mind, the darkest columns in these tables include adjectives (ADJ), adverbs (ADV), conjunctions (CONJ), nouns (NOUN), other (OTHER), verbs (VERB) and word order (WO) errors. It should be made clear however, that these categories mainly contain word choice errors, such as [*eat* → *consume*], and that morphological errors, such as [*eat* → *eating*], are variously subsumed under other categories. The results indicate that while systems are fairly adept at correcting morphological and function word errors, they struggle with content word errors. Content word errors require a deeper understanding of the text compared to morphological and function word errors. Such errors should not be ignored however, and ADJ, ADV, NOUN and VERB errors collectively account for over 10% of all errors, which is equal to the 3rd most frequent error type.

In terms of error types overall, UEDIN-MS was the most successful team and scored highest on 15/24 error types in the Restricted Track and 20/24 in the Low Resource Track. YDGEC meanwhile came 2nd in the Restricted Track, scoring highest on 3/24 error types, while a handful of other teams did best at 1 or 2 types. YDGEC is also notable for scoring much better at adjective and adverb errors than UEDIN-MS; it would be interesting to determine why. In contrast, UEDIN-MS performed significantly better on content word errors in the Low Resource Track than their nearest competitors, which suggests that their artificial data generation method might also be proficient at simulating content word errors.

Finally, the team that came 5th overall, Shuyao, came last in terms of orthography (ORTH) errors, even though they constitute the 5th most frequent error type. This not only indicates a straightforward way for them to improve system, but also demonstrates how an ERRANT error type analysis can help guide the system development process.

## A.7 All Metrics

As mentioned at the start of this section, we chose to use gold annotated references as the official references in the shared task even though all system hypotheses were annotated automatically by ERRANT. One consequence of this however, is that systems are unlikely to reach 100 $F_{0.5}$ even if they produce exactly the same corrected sentences as the references. This is because ERRANT computes scores in terms of edit overlap, yet automatic edit spans do not always match human edit spans; for example ERRANT will merge edits such as [$\epsilon$ → *has*] and [*eat* → *eaten*] into [*eat* → *has eaten*], but human annotators may choose to keep them separate. Consequently, although the edits ultimately produce the same correction, the automatic hypothesis does not match the gold reference and so the system is not rewarded. This explains why some teams found that submitting the official corrected development sentences to Codalab during the development phase only scored ∼86 $F_{0.5}$.

In this section, we additionally report system performance using automatic references instead of gold references. While it may seem unorthodox to use automatic references instead of gold references, the main advantage of this setting is that all the edits in the hypothesis and reference files are classified under exactly the same conditions. This not only means hypothesis edits are more likely to match the reference edits, but also that the official corrected sentences will score the maximum 100 $F_{0.5}$ on the development and test sets. Table 12 hence shows that the ERRANT $F_{0.5}$ scores of almost all teams in all tracks increased when compared against the automatic references, which indicates that systems are now rewarded for valid edits that were previously overlooked.

In addition to evaluating systems using gold and automatic references with ERRANT, we also evaluated systems using the other most popular metrics in GEC; namely MaxMatch (Dahlmeier and Ng, 2012), the I-measure (Felice and Briscoe, 2015), and GLEU (Napoles et al., 2015). The results, as well as how they affect each team's ranking, are also shown in Table 12. Note that the I-measure and GLEU are unaffected by the differences between gold and auto references and so are only reported once in this table.

Although we see that the rankings do change depending on the metric and type of reference, UEDIN-MS still came top in all settings in both the Restricted and Low Resource Tracks. While Kakao&Brain also consistently came second in almost all metrics, the exception was GLEU in the Restricted Track where they dropped to 5th. The overall GLEU rankings deviate significantly from the other metrics and also strongly correlate with recall. For example, ML@IITB, BLCU and TMU all ranked much better under GLEU, on account of their higher recall, while LAIX dropped from 3rd to 9th because their system emphasised precision. We additionally note that the range in scores for the top 19 teams in the Restricted Track was less than 7.5 using GLEU, but over 25 $F_{0.5}$ for both ERRANT and MaxMatch and 40 in terms of the I-measure. We thus conclude that GLEU is less discriminative than other metrics.

Finally, although MaxMatch $F_{0.5}$ scores tended to be higher than ERRANT $F_{0.5}$ scores in both the gold and auto reference settings, we note that MaxMatch exploits a dynamic alignment to artificially minimise the false positive rate and hence produces slightly inflated scores (Bryant et al., 2017). We also note that despite previous research that suggested MaxMatch correlates more strongly with human judgements than the I-measure (cf. Section 5), the I-measure still ranked the top 10 Restricted Track systems in exactly the same order as MaxMatch $F_{0.5}$. We hope that these results will encourage researchers to investigate further and perhaps develop better evaluation practices.

**Restricted**

| Teams | ADJ | ADJ FORM | ADV | CONJ | CONTR | DET | MORPH | NOUN | NOUN INFL | NOUN NUM | NOUN POSS | ORTH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UEDIN-MS | 43.48 | 83.33 | 49.41 | 48.67 | 84.75 | 75.67 | 79.31 | 41.17 | 91.95 | 79.92 | 83.68 | 82.10 |
| Kakao&Brain | 42.68 | 74.07 | 47.41 | 19.23 | 98.21 | 70.24 | 70.45 | 31.82 | 77.78 | 75.32 | 68.97 | 75.77 |
| LAIX | 46.05 | 54.05 | 45.11 | 16.67 | 76.92 | 70.07 | 74.16 | 34.09 | 81.52 | 67.40 | 63.32 | 73.02 |
| CAMB-CLED | 41.94 | 78.95 | 51.65 | 28.46 | 77.92 | 71.87 | 76.47 | 34.75 | 67.31 | 71.12 | 69.05 | 80.69 |
| Shuyao | 47.37 | 83.33 | 56.64 | 40.00 | 91.67 | 73.10 | 70.54 | 33.33 | 72.29 | 73.90 | 66.67 | 50.41 |
| YDGEC | 53.10 | 76.92 | 55.02 | 32.26 | 75.47 | 70.42 | 67.46 | 25.84 | 77.59 | 73.42 | 64.63 | 71.08 |
| ML@IITB | 19.90 | 53.57 | 46.04 | 58.14 | 68.97 | 72.53 | 63.62 | 17.73 | 23.62 | 72.52 | 68.63 | 67.29 |
| CAMB-CUED | 50.30 | 65.22 | 53.69 | 36.08 | 74.47 | 68.33 | 72.48 | 34.05 | 52.08 | 71.21 | 69.31 | 78.61 |
| AIP-Tohoku | 41.67 | 90.91 | 51.92 | 28.17 | 81.82 | 68.09 | 58.69 | 29.96 | 69.77 | 69.64 | 58.59 | 65.85 |
| UFAL | 43.48 | 74.07 | 50.00 | 32.79 | 83.33 | 63.23 | 72.29 | 25.24 | 60.61 | 65.02 | 52.63 | 75.68 |
| CVTE-NLP | 46.73 | 83.33 | 43.86 | 45.45 | 86.54 | 59.56 | 62.37 | 29.24 | 86.96 | 68.54 | 61.22 | 72.46 |
| BLCU | 50.00 | 83.33 | 44.12 | 29.70 | 61.64 | 64.30 | 65.53 | 22.29 | 68.42 | 66.69 | 58.14 | 75.63 |
| IBM | 28.30 | 66.67 | 0.00 | 0.00 | 65.22 | 57.64 | 51.37 | 12.58 | 0.00 | 62.19 | 24.75 | 53.35 |
| TMU | 24.62 | 58.14 | 32.29 | 39.82 | 79.37 | 61.65 | 63.69 | 22.32 | 72.92 | 60.53 | 73.30 | 74.76 |
| qiuwenbo | 38.14 | 62.50 | 43.15 | 16.13 | 54.05 | 53.78 | 57.32 | 23.32 | 86.96 | 64.57 | 45.00 | 70.18 |
| NLG-NTU | 12.82 | 41.67 | 34.74 | 36.04 | 70.00 | 53.09 | 49.38 | 9.38 | 58.82 | 55.78 | 60.13 | 69.15 |
| CAI | 19.48 | 45.45 | 31.25 | 28.46 | 90.16 | 49.31 | 54.01 | 14.60 | 58.82 | 52.21 | 64.71 | 70.28 |
| PKU | 35.97 | 62.50 | 34.19 | 25.32 | 81.40 | 59.79 | 58.46 | 13.00 | 60.00 | 63.66 | 40.54 | 69.94 |
| SolomonLab | 17.42 | 62.50 | 52.08 | 23.44 | 77.78 | 53.23 | 36.62 | 12.82 | 87.63 | 57.58 | 57.02 | 59.15 |
| Buffalo | 36.76 | 58.82 | 26.32 | 0.00 | 50.00 | 37.13 | 49.36 | 18.63 | 57.14 | 52.90 | 18.52 | 53.55 |
| Ramaiah | 3.26 | 55.56 | 12.86 | 3.40 | 41.67 | 23.10 | 33.86 | 0.80 | 0.00 | 40.61 | 17.24 | 58.60 |
| **Freq. (%)** | **1.05** | **0.18** | **1.45** | **0.75** | **0.32** | **10.41** | **2.50** | **2.89** | **0.28** | **4.07** | **0.93** | **8.03** |

| Teams | OTHER | PART | PREP | PRON | PUNCT | SPELL | VERB | VERB FORM | VERB INFL | VERB SVA | VERB TENSE | WO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UEDIN-MS | 45.59 | 66.90 | 71.81 | 68.47 | 67.87 | 82.71 | 59.27 | 79.52 | 97.22 | 86.74 | 66.20 | 54.27 |
| Kakao&Brain | 34.36 | 73.53 | 65.56 | 67.12 | 78.17 | 76.12 | 43.33 | 71.65 | 96.77 | 83.11 | 63.08 | 47.69 |
| LAIX | 23.99 | 68.42 | 62.85 | 62.99 | 75.66 | 72.82 | 30.30 | 75.80 | 86.21 | 78.95 | 56.92 | 47.32 |
| CAMB-CLED | 38.95 | 74.07 | 65.40 | 64.55 | 75.02 | 77.51 | 39.80 | 75.63 | 93.75 | 80.08 | 60.82 | 52.71 |
| Shuyao | 40.49 | 69.54 | 65.86 | 67.68 | 76.41 | 77.22 | 53.52 | 78.02 | 97.22 | 80.37 | 60.55 | 59.47 |
| YDGEC | 37.13 | 75.76 | 65.53 | 54.69 | 70.01 | 77.37 | 49.28 | 77.11 | 100.00 | 78.37 | 62.04 | 50.65 |
| ML@IITB | 31.75 | 65.84 | 67.35 | 62.86 | 75.89 | 67.93 | 49.19 | 75.93 | 86.96 | 84.40 | 58.82 | 60.14 |
| CAMB-CUED | 35.50 | 72.25 | 59.85 | 61.57 | 72.64 | 73.44 | 40.13 | 73.43 | 89.29 | 79.21 | 55.60 | 52.56 |
| AIP-Tohoku | 34.77 | 69.67 | 60.59 | 51.17 | 70.42 | 70.77 | 42.19 | 71.51 | 62.50 | 75.55 | 54.39 | 46.75 |
| UFAL | 29.79 | 54.35 | 55.82 | 57.74 | 70.44 | 63.32 | 44.75 | 74.36 | 71.43 | 77.81 | 51.48 | 47.52 |
| CVTE-NLP | 24.79 | 67.01 | 51.16 | 54.57 | 64.36 | 75.73 | 40.60 | 69.26 | 94.59 | 72.97 | 49.13 | 47.62 |
| BLCU | 30.36 | 58.06 | 59.17 | 48.11 | 66.72 | 66.39 | 45.57 | 71.29 | 96.77 | 76.06 | 50.66 | 61.92 |
| IBM | 15.10 | 51.02 | 48.95 | 43.40 | 66.81 | 66.80 | 21.38 | 62.50 | 0.00 | 70.82 | 51.66 | 36.89 |
| TMU | 23.84 | 52.88 | 54.62 | 45.32 | 70.83 | 63.17 | 32.94 | 63.64 | 94.59 | 73.85 | 49.16 | 43.00 |
| qiuwenbo | 22.16 | 58.82 | 41.19 | 52.63 | 48.94 | 74.36 | 30.94 | 66.19 | 86.21 | 71.68 | 44.19 | 44.60 |
| NLG-NTU | 16.41 | 62.50 | 45.43 | 47.39 | 62.23 | 53.64 | 32.44 | 60.40 | 73.53 | 66.06 | 42.04 | 41.81 |
| CAI | 17.98 | 48.00 | 43.71 | 42.24 | 60.57 | 56.14 | 25.22 | 56.58 | 94.59 | 66.36 | 33.83 | 28.07 |
| PKU | 14.73 | 63.73 | 49.96 | 52.56 | 61.46 | 60.00 | 27.23 | 69.90 | 80.00 | 71.43 | 44.38 | 45.70 |
| SolomonLab | 16.20 | 62.91 | 48.73 | 37.38 | 26.38 | 66.67 | 29.32 | 50.71 | 89.29 | 59.27 | 38.80 | 39.80 |
| Buffalo | 7.68 | 47.62 | 21.01 | 31.03 | 30.17 | 50.00 | 11.47 | 65.32 | 38.46 | 65.29 | 34.05 | 12.05 |
| Ramaiah | 0.73 | 38.46 | 21.90 | 22.85 | 51.28 | 5.63 | 3.95 | 48.78 | 58.82 | 52.97 | 32.32 | 34.38 |
| **Freq. (%)** | **15.69** | **0.49** | **8.33** | **2.45** | **16.73** | **4.63** | **5.09** | **3.10** | **0.12** | **2.28** | **5.43** | **1.40** |

Table 10: Main error type ERRANT $F_{0.5}$ scores for each team in the Restricted Track. Darker red indicates a lower score. The percent frequency of each type in the test set is also shown.

**Unrestricted**

| Teams | ADJ | ADJ FORM | ADV | CONJ | CONTR | DET | MORPH | NOUN | NOUN INFL | NOUN NUM | NOUN POSS | ORTH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LAIX | 46.05 | 54.05 | 45.11 | 16.67 | 76.92 | 70.07 | 74.16 | 34.09 | 81.52 | 67.40 | 63.32 | 73.02 |
| AIP-Tohoku | 53.96 | 83.33 | 49.08 | 51.14 | 92.31 | 68.74 | 68.38 | 38.37 | 89.04 | 74.74 | 71.43 | 71.02 |
| UFAL | 50.96 | 69.77 | 46.03 | 37.74 | 82.19 | 66.86 | 70.72 | 37.91 | 78.57 | 71.43 | 76.04 | 81.37 |
| BLCU | 50.76 | 78.43 | 42.57 | 43.62 | 68.42 | 59.67 | 66.39 | 33.52 | 53.85 | 67.11 | 59.81 | 75.79 |
| Aparecium | 37.74 | 43.48 | 39.53 | 32.61 | 30.77 | 55.87 | 52.42 | 18.78 | 44.44 | 65.93 | 50.56 | 70.69 |
| Buffalo | 10.87 | 53.57 | 37.04 | 0.00 | 66.67 | 43.26 | 56.16 | 13.23 | 49.02 | 53.15 | 33.33 | 51.75 |
| Ramaiah | 1.98 | 9.26 | 11.59 | 0.00 | 52.63 | 30.30 | 30.75 | 1.20 | 0.00 | 41.30 | 9.90 | 54.82 |
| **Freq. (%)** | **1.05** | **0.18** | **1.45** | **0.75** | **0.32** | **10.41** | **2.50** | **2.89** | **0.28** | **4.07** | **0.93** | **8.03** |

| Teams | OTHER | PART | PREP | PRON | PUNCT | SPELL | VERB | VERB FORM | VERB INFL | VERB SVA | VERB TENSE | WO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LAIX | 23.99 | 68.42 | 62.85 | 62.99 | 75.66 | 72.82 | 30.30 | 75.80 | 86.21 | 78.95 | 56.92 | 47.32 |
| AIP-Tohoku | 44.05 | 71.97 | 62.37 | 67.71 | 72.34 | 79.40 | 45.58 | 76.09 | 89.29 | 77.31 | 59.75 | 57.18 |
| UFAL | 36.50 | 75.76 | 61.79 | 57.29 | 70.49 | 84.80 | 49.45 | 72.61 | 89.29 | 78.91 | 59.83 | 43.41 |
| BLCU | 34.98 | 63.16 | 58.68 | 61.15 | 65.86 | 77.81 | 43.27 | 70.85 | 97.22 | 74.70 | 55.41 | 61.29 |
| Aparecium | 18.63 | 64.71 | 47.44 | 49.85 | 57.17 | 61.71 | 31.20 | 68.29 | 93.75 | 75.04 | 44.64 | 34.81 |
| Buffalo | 10.70 | 52.63 | 30.16 | 34.29 | 31.54 | 50.32 | 17.39 | 72.44 | 38.46 | 71.78 | 35.50 | 32.00 |
| Ramaiah | 0.84 | 32.26 | 31.67 | 26.47 | 55.82 | 4.23 | 3.88 | 47.82 | 41.67 | 47.17 | 20.71 | 32.89 |
| **Freq. (%)** | **15.69** | **0.49** | **8.33** | **2.45** | **16.73** | **4.63** | **5.09** | **3.10** | **0.12** | **2.28** | **5.43** | **1.40** |

**Low Resource**

| Teams | ADJ | ADJ FORM | ADV | CONJ | CONTR | DET | MORPH | NOUN | NOUN INFL | NOUN NUM | NOUN POSS | ORTH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UEDIN-MS | 46.39 | 83.33 | 39.39 | 25.42 | 51.72 | 64.01 | 72.25 | 41.13 | 92.59 | 77.23 | 79.21 | 79.23 |
| Kakao&Brain | 0.00 | 50.00 | 6.10 | 0.00 | 44.64 | 58.17 | 52.40 | 16.51 | 75.58 | 56.61 | 18.29 | 65.14 |
| LAIX | 0.00 | 31.25 | 9.43 | 0.00 | 0.00 | 51.35 | 61.71 | 19.42 | 80.00 | 57.36 | 35.85 | 51.16 |
| CAMB-CUED | 0.00 | 17.86 | 0.00 | 19.13 | 35.71 | 40.91 | 37.18 | 13.51 | 93.02 | 59.71 | 47.39 | 73.31 |
| UFAL | 32.11 | 33.33 | 24.00 | 10.64 | 7.69 | 26.20 | 48.28 | 30.49 | 93.41 | 66.33 | 64.52 | 70.56 |
| Siteimprove | 8.20 | 0.00 | 9.80 | 2.48 | 0.00 | 18.63 | 35.71 | 20.83 | 40.00 | 47.18 | 0.00 | 4.59 |
| WebSpellChecker | 9.98 | 0.00 | 16.47 | 8.33 | 33.33 | 54.43 | 38.67 | 10.58 | 37.04 | 56.07 | 49.50 | 67.47 |
| TMU | 1.66 | 36.59 | 6.99 | 19.44 | 0.00 | 26.50 | 24.75 | 1.82 | 32.05 | 38.71 | 10.20 | 45.95 |
| Buffalo | 17.54 | 0.00 | 22.47 | 0.00 | 21.43 | 10.64 | 23.29 | 7.97 | 22.73 | 19.59 | 17.24 | 49.28 |
| **Freq. (%)** | **1.05** | **0.18** | **1.45** | **0.75** | **0.32** | **10.41** | **2.50** | **2.89** | **0.28** | **4.07** | **0.93** | **8.03** |

| Teams | OTHER | PART | PREP | PRON | PUNCT | SPELL | VERB | VERB FORM | VERB INFL | VERB SVA | VERB TENSE | WO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UEDIN-MS | 38.51 | 73.53 | 62.01 | 62.26 | 62.85 | 84.09 | 49.12 | 78.17 | 97.22 | 76.59 | 50.56 | 29.97 |
| Kakao&Brain | 17.61 | 51.02 | 47.22 | 49.00 | 74.64 | 73.82 | 20.72 | 61.11 | 96.77 | 74.35 | 47.36 | 11.81 |
| LAIX | 4.16 | 32.79 | 41.18 | 11.63 | 64.42 | 60.64 | 0.00 | 55.18 | 0.00 | 67.31 | 0.00 | 0.00 |
| CAMB-CUED | 9.09 | 52.45 | 50.45 | 22.14 | 51.88 | 68.49 | 4.89 | 60.22 | 97.22 | 85.25 | 39.21 | 4.03 |
| UFAL | 21.73 | 42.86 | 27.03 | 24.19 | 33.17 | 80.25 | 27.86 | 58.46 | 93.75 | 72.18 | 21.02 | 19.05 |
| Siteimprove | 13.31 | 37.23 | 39.58 | 30.63 | 50.88 | 76.22 | 8.23 | 48.55 | 96.77 | 76.06 | 23.10 | 0.00 |
| WebSpellChecker | 8.38 | 41.67 | 37.97 | 33.42 | 66.74 | 42.91 | 17.89 | 54.26 | 33.33 | 71.73 | 35.21 | 42.15 |
| TMU | 2.51 | 30.00 | 18.12 | 19.30 | 46.16 | 65.50 | 9.27 | 28.43 | 75.00 | 30.22 | 14.96 | 18.07 |
| Buffalo | 5.19 | 29.41 | 11.74 | 14.04 | 36.23 | 6.35 | 7.50 | 8.17 | 66.67 | 12.82 | 6.24 | 28.00 |
| **Freq. (%)** | **15.69** | **0.49** | **8.33** | **2.45** | **16.73** | **4.63** | **5.09** | **3.10** | **0.12** | **2.28** | **5.43** | **1.40** |

Table 11: Main error type ERRANT $F_{0.5}$ scores for each team in the Unrestricted and Low Resource Track. Darker red indicates a lower score. The percent frequency of each type in the test set is also shown.

| | **ERRANT** | | | | | | **MaxMatch** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Restricted** | **Gold** | | | | **Auto** | | **Gold** | | **Auto** | | | | | |
| **Teams** | **P** | **R** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | **I** | **#** | **GLEU** | **#** |
| UEDIN-MS | 77.87 | 62.29 | 69.47 | 1 | 74.16 | 1 | 76.48 | 1 | 76.62 | 1 | 38.92 | 1 | 77.93 | 1 |
| Kakao&Brain | 80.18 | 53.28 | 69.00 | 2 | 72.83 | 2 | 74.09 | 2 | 74.17 | 2 | 36.84 | 2 | 75.87 | 5 |
| LAIX | 77.03 | 50.19 | 66.78 | 3 | 69.59 | 5 | 70.78 | 7 | 70.79 | 7 | 28.20 | 7 | 74.33 | 9 |
| CAMB-CLED | 74.59 | 56.53 | 66.75 | 4 | 70.11 | 3 | 72.51 | 3 | 72.48 | 3 | 34.10 | 3 | 76.62 | 3 |
| Shuyao | 74.41 | 56.31 | 66.61 | 5 | 69.91 | 4 | 72.22 | 4 | 72.37 | 4 | 33.22 | 4 | 76.55 | 4 |
| YDGEC | 74.50 | 54.49 | 65.83 | 6 | 69.41 | 6 | 71.60 | 6 | 71.20 | 6 | 29.21 | 6 | 75.39 | 7 |
| ML@IITB | 69.69 | 63.29 | 64.73 | 7 | 68.30 | 7 | 71.97 | 5 | 71.75 | 5 | 30.75 | 5 | 77.89 | 2 |
| CAMB-CUED | 71.49 | 55.63 | 63.72 | 8 | 67.63 | 8 | 70.37 | 8 | 70.44 | 8 | 26.37 | 8 | 75.82 | 6 |
| AIP-Tohoku | 72.79 | 43.05 | 60.97 | 9 | 63.95 | 9 | 65.95 | 9 | 65.84 | 9 | 19.22 | 9 | 73.16 | 11 |
| UFAL | 71.56 | 41.21 | 59.39 | 10 | 62.37 | 10 | 65.70 | 10 | 65.19 | 10 | 17.46 | 10 | 72.79 | 12 |
| CVTE-NLP | 72.12 | 39.12 | 59.22 | 11 | 61.71 | 12 | 63.04 | 12 | 63.17 | 12 | 16.71 | 11 | 72.51 | 13 |
| BLCU | 65.11 | 52.54 | 58.62 | 12 | 62.14 | 11 | 64.82 | 11 | 65.05 | 11 | 13.04 | 12 | 74.33 | 8 |
| IBM | 66.19 | 37.45 | 55.74 | 13 | 57.38 | 13 | 59.47 | 14 | 58.79 | 14 | 8.84 | 14 | 71.48 | 15 |
| TMU | 57.69 | 53.15 | 53.45 | 14 | 56.72 | 14 | 61.44 | 13 | 61.60 | 13 | -0.54 | 17 | 73.96 | 10 |
| qiuwenbo | 66.56 | 32.84 | 52.80 | 15 | 55.22 | 15 | 57.70 | 15 | 57.22 | 15 | 8.94 | 13 | 71.30 | 16 |
| LG-NTU | 52.54 | 39.20 | 46.77 | 16 | 49.19 | 17 | 53.38 | 17 | 53.15 | 17 | -1.45 | 18 | 71.13 | 17 |
| CAI | 51.49 | 42.61 | 46.69 | 17 | 49.43 | 16 | 53.68 | 16 | 53.56 | 16 | -1.49 | 19 | 71.68 | 14 |
| PKU | 54.84 | 32.17 | 46.64 | 18 | 48.06 | 18 | 52.84 | 18 | 52.30 | 18 | -0.32 | 15 | 71.06 | 18 |
| SolomonLab | 47.05 | 39.69 | 43.73 | 19 | 45.37 | 19 | 50.00 | 19 | 50.40 | 19 | -3.50 | 20 | 70.56 | 19 |
| Buffalo | 65.09 | 15.08 | 39.06 | 20 | 39.14 | 20 | 40.95 | 20 | 40.13 | 20 | -0.32 | 15 | 68.32 | 20 |
| Ramaiah | 10.29 | 19.04 | 10.83 | 21 | 11.33 | 21 | 18.68 | 21 | 18.49 | 21 | -21.78 | 21 | 56.31 | 21 |

| | **ERRANT** | | | | | | **MaxMatch** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Unrestricted** | **Gold** | | | | **Auto** | | **Gold** | | **Auto** | | | | | |
| **Teams** | **P** | **R** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | **I** | **#** | **GLEU** | **#** |
| LAIX | 77.03 | 50.19 | 66.78 | 1 | 69.59 | 1 | 70.78 | 3 | 70.79 | 3 | 28.20 | 3 | 74.33 | 3 |
| AIP-Tohoku | 75.45 | 52.59 | 65.57 | 2 | 69.41 | 2 | 70.93 | 2 | 70.98 | 2 | 28.65 | 2 | 74.83 | 2 |
| UFAL | 73.35 | 55.14 | 64.55 | 3 | 68.81 | 3 | 71.74 | 1 | 71.48 | 1 | 29.65 | 1 | 75.83 | 1 |
| BLCU | 64.56 | 58.17 | 59.50 | 4 | 63.17 | 4 | 65.42 | 4 | 65.74 | 4 | 7.08 | 4 | 74.11 | 4 |
| Aparecium | 61.87 | 36.09 | 52.76 | 5 | 54.14 | 5 | 55.61 | 5 | 55.80 | 5 | 5.57 | 5 | 71.96 | 5 |
| Buffalo | 66.17 | 17.19 | 42.33 | 6 | 42.15 | 6 | 44.33 | 6 | 43.09 | 6 | 4.25 | 6 | 68.77 | 6 |
| Ramaiah | 13.09 | 24.94 | 13.98 | 7 | 14.46 | 7 | 22.10 | 7 | 22.00 | 7 | -20.13 | 7 | 57.50 | 7 |

| | **ERRANT** | | | | | | **MaxMatch** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Low Resource** | **Gold** | | | | **Auto** | | **Gold** | | **Auto** | | | | | |
| **Teams** | **P** | **R** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | $\mathbf{F_{0.5}}$ | **#** | **I** | **#** | **GLEU** | **#** |
| UEDIN-MS | 72.97 | 47.86 | 64.24 | 1 | 66.04 | 1 | 67.34 | 1 | 67.39 | 1 | 16.06 | 1 | 74.30 | 1 |
| Kakao&Brain | 65.75 | 46.73 | 58.80 | 2 | 60.80 | 2 | 63.51 | 2 | 63.04 | 2 | 15.23 | 2 | 73.98 | 2 |
| LAIX | 63.86 | 30.93 | 51.81 | 3 | 52.65 | 3 | 53.84 | 4 | 53.64 | 4 | 4.73 | 3 | 70.76 | 4 |
| CAMB-CUED | 56.77 | 37.42 | 50.88 | 4 | 51.45 | 4 | 54.32 | 3 | 54.09 | 3 | -0.16 | 4 | 71.86 | 3 |
| UFAL | 52.82 | 29.23 | 44.13 | 5 | 45.48 | 5 | 49.28 | 5 | 49.34 | 5 | -3.24 | 7 | 69.39 | 6 |
| Siteimprove | 45.34 | 28.26 | 40.17 | 6 | 40.45 | 7 | 42.59 | 7 | 42.99 | 7 | -1.48 | 5 | 69.29 | 7 |
| WebSpellChecker | 40.79 | 44.08 | 39.75 | 7 | 41.41 | 6 | 48.88 | 6 | 48.08 | 6 | -4.58 | 8 | 69.76 | 5 |
| TMU | 28.21 | 31.61 | 28.31 | 8 | 28.83 | 8 | 32.09 | 8 | 32.20 | 8 | -6.98 | 9 | 65.50 | 9 |
| Buffalo | 25.87 | 10.37 | 20.73 | 9 | 19.92 | 9 | 22.55 | 9 | 21.63 | 9 | -2.39 | 6 | 65.82 | 8 |

Table 12: ERRANT $F_{0.5}$ scores on the official gold references are compared against automatic references and other popular metrics. The differences in how these metrics would rank each team are also shown, where a darker red indicates a lower rank.