Grammatical Error Correction: A Survey of the State of the Art

Christopher Bryant
ALTA Institute,
University of Cambridge

Zheng Yuan
Department of Informatics,
King's College London

Muhammad Reza Qorib
National University of Singapore

Hannan Cao
National University of Singapore

Hwee Tou Ng
National University of Singapore

Ted Briscoe
ALTA Institute,
University of Cambridge

Grammatical Error Correction (GEC) is the task of automatically detecting and correcting errors in text. The task not only includes the correction of grammatical errors, such as missing prepositions and mismatched subject-verb agreement, but also orthographic and semantic errors, such as misspellings and word choice errors respectively. The field has seen significant progress in the last decade, motivated in part by a series of five shared tasks, which drove the development of rule-based methods, statistical classifiers, statistical machine translation, and finally neural machine translation systems which represent the current dominant state of the art. In this survey paper, we condense the field into a single article and first outline some of the linguistic challenges of the task, introduce the most popular datasets that are available to researchers (for both English and other languages), and summarise the various methods and techniques that have been developed with a particular focus on artificial error generation. We next describe the many different approaches to evaluation as well as concerns surrounding metric reliability, especially in relation to subjective human judgements, before concluding with an overview of recent progress and suggestions for future work and remaining challenges. We hope that this survey will serve as comprehensive resource for researchers who are new to the field or who want to be kept apprised of recent developments.

1. Introduction

Writing is a learnt skill that is particularly challenging for non-native language users. We all make occasional mistakes with punctuation, spelling and minor infelicities of word choice in our native language, but non-native writers often also struggle to create grammatical and comprehensible texts. Research in the field of Natural Language Processing (NLP) has addressed the problem of 'ill-formed input' at least since the 1980s because downstream parsing of text usually collapsed unless input was grammatical (Kwasny and Sondheimer 1981; Jensen et al. 1983). However, useful applications able to significantly assist non-native writers only began to appear in the 2000s, such as ETS's Criterion (Burstein, Chodorow, and Leacock 2003) and Microsoft's ESL Assistant (Leacock, Gamon, and Brockett 2009). These systems were largely based on hand-coded

'mal-rules' applied to the output from robust parsers which suggested corrections for errors.

Around the same time, researchers began exploring more data-driven approaches using supervised machine learning models built from annotated corpora of errorful text with exemplary corrections (Brockett, Dolan, and Gamon 2006; De Felice and Pulman 2008; Rozovskaya and Roth 2010b; Tetreault, Foster, and Chodorow 2010; Dahlmeier and Ng 2011b). The Helping Our Own (HOO) shared task (Dale, Anisimoff, and Narroway 2012), which attracted 14 research groups to compete and report their results on correcting English determiner and preposition choice errors using the First Certificate in English (FCE) corpus (Yannakoudakis, Briscoe, and Medlock 2011), marked with hindsight the turning point from rule-based to data-driven methods as well as burgeoning interest in the task. Leacock et al. (2014) subsequently published a book length survey summarising progress in the field up to this point.

The next decade has seen three further expanded shared tasks and an explosion of research and publications, both from participants in these competitions and others benchmarking their systems against the released test sets. Performance has increased roughly three-fold, and today, most state-of-the-art systems treat the task as one of 'translation' from errorful to corrected text, including the latest system deployed in Google Docs and Gmail (Hoskere 2019). Recently, Wang et al. (2021) provided another detailed survey of work on grammatical error correction summarising most work published since Leacock et al. (2014). In this article, we provide a more in-depth focus on very recent deep learning based approaches to the task as well as a more detailed discussion of the nature of the task, its evaluation, and other remaining challenges (such as multilingual GEC) in order to better equip researchers with the insights required to be able to contribute to further progress.

1.1 The Task

The definition of a grammatical error is surprisingly difficult. Some types of spelling errors (such as *accomodation* with a single *m*) are about equally distributed between native and non-native writers and have no grammatical reflexes, so could be reasonably excluded. Others, such as *he eated*, are boundary cases as they result from over-regularisation of morphology, whilst *he would eated* is clearly ungrammatical in the context of a modal auxiliary verb. At the interpretative boundary, infelicitous discourse organisation, such as *Kim fell. Sandy pushed him.* where the intention is to explain why Kim fell, is not obviously a grammatical error *per se* but nevertheless can be 'corrected' via a tense change (*Sandy had pushed him.*) as opposed to a reordering of the sentences. Other tense changes which can span sentences appear more grammatical, such as *Kim will make Sandy a sandwich. Sandy ate it.*, as the discourse is incoherent and correction will require a tense change in one or other sentence.

In practice, the task has increasingly been defined in terms of what corrections are annotated in corpora used for the shared tasks. These use a variety of annotation schemes but all tend to adopt minimal modifications of errorful texts to create error-free text with the same perceived meaning. Other sources of annotated data, such as that sourced from the online language learning platform Lang-8 (Mizumoto et al. 2012; Tajiri, Komachi, and Matsumoto 2012), often contain much more extensive rewrites of entire paragraphs of text. Given this resource-derived definition of the task, systems are evaluated on their ability to correct all kinds of mistakes in text, including spelling and discourse level errors that have no or little grammatical reflex. The term 'Grammatical' Error Correction is thus something of a misnomer, but is nevertheless now commonly

Type	Error	Correction
Preposition	I sat in the talk	I sat in on the talk
Morphology	dreamed	dreamt
Determiner	I like the ice cream	I like ice cream
Tense/Aspect	I like kiss you	I like kissing you
Agreement	She likes him and kiss him	She likes him and kisses him
Syntax	I have not the book	I do not have the book
Punctuation	We met they talked and left	We met, they talked and left
Unidiomatic	We had a big conversation	We had a long conversation
Multiple	I sea the see from the seasoar	I saw the sea from the seesaw

Table 1
Example error types

understood to encompass errors that are not always strictly grammatical in nature. A more descriptive term is *Language Error Correction*.

Table 1 provides a small sample of (constructed) examples that illustrate the range of errors to be corrected and some of the issues that arise with the precise definition and evaluation of the task. Errors can be classified into three broad categories: **replacement** errors, such as *dreamed* for *dreamt* in the second example; **omission** errors, such as *on* in the first example; and **insertion** errors, such as *the* in the third example. Some errors are complex in the sense that their correction requires a sequence of replacement, omission or insertion steps to correct, as with the syntax example. Sentences may also contain multiple distinct errors that require a sequence of corrections, as in the multiple example. Both the classification and specification of correction steps for errors can be and has been achieved using different schemes and approaches. For instance, correction of the syntax example involves transposing two adjacent words so we could introduce a fourth broad class and correction step of transposition (word order). All extant annotation schemes break these broad classes down into further subclasses based on the part-of-speech of the words involved, and perceived morphological, lexical, syntactic, semantic or pragmatic source of the error. The schemes vary in the number of such distinctions, ranging from just over two dozen (NUCLE: (Dahlmeier, Ng, and Wu 2013)) to almost a hundred (CLC: (Nicholls 2003)). The schemes also identify different error spans in source sentences and thus suggest different sets of edit operations to obtain the suggested corrections. For instance, the agreement error example might be annotated as *She likes him and* [*kiss* $\rightarrow$ *kisses*] *him* at the token level or simply [ϵ $\rightarrow$ *es*] at the character level. These differing annotation decisions affected the evaluation of system performance in artefactual ways, so a two-stage automatic standardisation process was developed, ERRANT (Felice, Bryant, and Briscoe 2016; Bryant, Felice, and Briscoe 2017), which maps parallel errorful and corrected sentence pairs to a single annotation scheme using a linguistically-enhanced alignment algorithm and series of error type classification rules. This scheme uses 25 main error type categories, based primarily on part-of-speech and morphology, which are further subdivided into missing (omission), unnecessary (insertion) and replacement errors. This approach allows consistent automated training and evaluation of systems on any or all parallel corpora as well as supporting a more fined-grained analysis of the strengths and weaknesses of systems in terms of different error types.

Ultimately however, the correction of errors requires an understanding of the communicative intention of the writer. For instance, the determiner example in Table 1 implicitly assumes a 'neutral' context where the intent is to make a statement about generic ice-cream rather than a specific instance. In a context where, say, a specific ice-cream dessert is being compared to an alternative dessert, then the determiner is felicitous. Similarly the preposition omission error might not be an error if the writer is describing a context in which a talk was oversubscribed and many attendees had to stand because of a lack of seats. Though annotators will most likely take both the context and perceived writer's intention into account when identifying errors, GEC itself is instead often framed as an isolated sentence-based task that ignores the wider context. This can introduce noise in the task in that errorful sequences in context may appear correct in isolation out of context. A related issue is that correction may not only depend on communicative intent, but also factors such as dialect and genre. For example, correcting *dreamed* to *dreamt* may be appropriate if the target is British English, but incorrect for American English.

A larger issue arises with differing possibilities for correction. For example, correcting the tense/aspect example to *kissing* or *to kiss* in the context of *likes* seems equally correct. However, few existing corpora provide more than one possibility which means the true performance of systems is often underestimated. However, the same two corrections are not equally correct as complements of a verb such as *try* depending on whether the context implies that a kissing event occurred or not. The issue of multiple possible corrections arises with many if not most examples: for instance *I haven't the book*, *We met them, talked and left*, *We had an important conversation*, *The sea I see from the seesaw (is calm)* are all plausible alternative corrections for some of the examples in Table 1. For this reason, several of the shared tasks have also evaluated performance on grammatical error detection, as this is valuable in some applications. Recently, some work has explored treating the GEC task as one of document-level correction (e.g. Chollampatt, Wang, and Ng (2019); Yuan and Bryant (2021)) which, in principle, could ameliorate some of these issues but is currently hampered by a lack of appropriately structured corpora.

1.2 Survey Structure

We organise the remainder of this survey according to Table 2:

2. Data

Like most tasks in NLP, the cornerstone of modern GEC systems is data. State-of-the-art neural models depend on millions or billions of words and the quality of this data is paramount to model success. Collecting high quality annotated data is a slow and laborious process however, and there are fewer resources available in GEC than other fields such as machine translation. This section hence first outlines some key considerations of data collection in GEC and highlights the importance of robust annotation guidelines. It next introduces the most commonly used corpora in English, as well as some less commonly used corpora, before concluding with GEC corpora for other languages. Artificial data has also become a popular topic in recent years, but this section only covers human annotated data; artificial data will be covered in Section 5.

	Subject	Topics
Section 2	Datasets	Data collection and annotation, benchmark English datasets, other English datasets, non-English datasets
Section 3	Approaches	Classifiers, statistical machine translation, neural machine translation, edit-based approaches, language models and low-resource systems
Section 4	Additional Techniques	Reranking, ensembling and system combination, multi-task learning, custom inference methods, contextual GEC, Generative Adversarial Networks (GANs)
Section 5	Artificial Data	Rule-based noise injection, probabilistic error patterns, back-translation, round-trip translation
Section 6	Evaluation	Benchmark metrics, reference-based metrics, reference-less metrics, metric reliability and human judgements, common experimental settings
Section 7	System Comparison	Recent state-of-the-art systems
Section 8	Future Challenges	Domain generalisation, personalised systems, feedback comment generation, model interpretability, semantic errors, contextual GEC, system combination, training data selection, unsupervised approaches, multilingual GEC, spoken GEC, improved evaluation
Section 9	Conclusion	-

Table 2
Survey structure

2.1 Annotation Challenges

As mentioned in Section 1.1, the notion of a grammatical error is hard to define as different errors may have different scope (e.g. local vs. contextual), complexity (e.g. orthographic vs. semantic) and corrections (e.g. [*this books* → *this book*] vs. [*this books* → *these books*]. Human annotation is thus an extremely cognitively demanding task and so clear annotation guidelines are a crucial component of dataset quality. This section briefly outlines three important aspects of data collection: Minimal vs. Fluent Corrections, Annotation Consistency, and Preprocessing Challenges.

Minimal vs. Fluent Corrections. Most GEC corpora have been annotated on the principle of *minimal corrections*, i.e. annotators should make the minimum number of changes to make a text grammatical. Sakaguchi et al. (2016) argue, however, that this can often lead to corrections that sound unnatural, and so it would be better to annotate corpora on the principle of *fluent corrections* instead. Consider the following example:

Original	I want explain to you some interesting part from my experience.
Minimal	I want <u>to</u> explain to you some interesting <u>parts of</u> my experience.
Fluent	I want <u>to tell you about</u> some interesting <u>parts of</u> my experience.

While the minimal correction primarily inserts a missing infinitival *to* before *explain* to make the sentence grammatical, the fluent correction also changes *explain* to *tell you about* because it is more idiomatic to tell someone about an experience rather than explain an experience.

One of the main challenges of this distinction, however, is that it is very difficult to draw a line between what constitutes a minimal correction and what constitutes a fluent correction. This is because minimal corrections (e.g. missing determiners) are a subset of fluent corrections, and so there cannot be fluent corrections without minimal corrections. It is also the case that minimal corrections are typically easier to make than

fluent corrections (for both humans and machines), although it is undeniable that fluent corrections are the more desirable outcome. Ultimately, although it is very difficult to precisely define a fluent correction, annotation guidelines should nevertheless attempt to make clear the extent to which annotators are expected to edit.

Annotation Consistency. A significant challenge of human annotation is that corrections are subjective and there is often more than one way to correct a sentence (Bryant and Ng 2015; Choshen and Abend 2018b). It is nevertheless important that annotators attempt to be consistent in their judgements, especially if they are explicitly annotating edit spans. For example the edit [*has eating* → *was eaten*] can also be represented as [*has* → *was*] and [*eating* → *eaten*], and this choice not only affects data exploration and analysis, but can also have an impact on edit-based evaluation. Similarly, the edit [*the informations* → *information*] can also be represented as [*the* → ϵ] and [*informations* → *information*], but the latter may be more intuitive because it represents two independent edits of clearly distinct types. Explicit error type classification is thus another important aspect of annotator consistency, as an error type framework (if any) not only increases the cognitive burden on the annotator, but also might influence an annotator towards a particular correction given the error types that are available (Sakaguchi et al. 2016). Ultimately, if annotators are tasked with explicitly defining the edits they make to correct a sentence, annotator guidelines must clearly define the notion of an edit.

Preprocessing Challenges. While human annotators are trained to correct natural text, GEC systems are typically trained to correct word tokenised sentences (mainly for evaluation purposes). This mismatch means human annotations typically undergo several preprocessing steps in order to produce the desired output format (Bryant and Felice 2016). The first of these transformations involves converting character-level edits to token-level edits. While this is often straightforward, it can sometimes be the case that a human-annotated character span does not map to a complete token; e.g. [*ing* → *ed*] to denote the edit [*dancing* → *danced*]. Although such cases can often (but not always) be resolved automatically, e.g., by expanding the character spans of the edit or calculating token alignment, they can also be reduced by training annotators to explicitly annotate longer spans rather than sub-words.

The second transformation involves sentence tokenisation, which is potentially more complex given human edits may change sentence boundaries; e.g. [*A. B, C.* → *A, B. C.*]. Sentences are nevertheless typically tokenised based solely on the original text, with the acknowledgement that some may be sentence fragments (to be joined with the following sentence) and that edits which cross sentence boundaries are ignored (e.g. [*. Because* → *, because*]. It is worth noting that this issue only affects sentence-based GEC systems (the vast majority) but paragraph or document-based systems are unaffected.

2.2 English Datasets

A small number of English GEC datasets have become popular for training and testing GEC systems, mostly as a result of shared tasks.[1] This section introduces them as well as other less popular datasets for English (Table 3). We acknowledge that this is by no means an exhaustive list, but highlight datasets that have gained some traction in the last few years.

1 `https://www.cl.cam.ac.uk/research/nl/bea2019st/#data`

Corpus	Use	Sents	Toks	Refs	Edit Spans	Error Types	Level	Domain
FCE	Train	28.3k	454k	1	✓	71	B1-B2	Exams
	Dev	2.2k	34.7k	1	✓	71	B1-B2	Exams
	Test	2.7k	41.9k	1	✓	71	B1-B2	Exams
NUCLE	Train	57.1k	1.16m	1	✓	28	C1	Essays
CoNLL-2013	Dev/Test	1.4k	29.2k	1	✓	28	C1	Essays
CoNLL-2014	Test	1.3k	30.1k	2-18	✓	28	C1	Essays
Lang-8	Train	1.03m	11.8m	1-8	✗	0	A1-C2?	Web
JFLEG	Dev	754	14.0k	4	✗	0	A1-C2?	Exams
	Test	747	14.1k	4	✗	0	A1-C2?	Exams
W&I+	Train	34.3k	628k	1	✓	55	A1-C2	Exams
LOCNESS	Dev	4.4k	87.0k	1	✓	55	A1-Native	Exams, Essays
(BEA-2019)	Test	4.5k	85.7k	5	✓	55	A1-Native	Exams, Essays
CLC	Train	1.96m	29.1m	1	✓	77	A1-C2	Exams
EFCamDat	Train	4.60m	56.8m	1	✓	25	A1-C2	Exams
WikEd	Train	28.5m	626m	1	✗	0	Native	Wiki
AESW	Train	1.20m	28.4m	1	✓	0	C1-Native	Science
	Dev	148k	3.51m	1	✓	0	C1-Native	Science
	Test	144k	3.45m	1	✓	0	C1-Native	Science
GMEG	Dev	2.9k	60.9k	4	✗	0	B1-B2, Native	Exams, Web, Wiki
	Test	2.9k	61.5k	4	✗	0	B1-B2, Native	Exams, Web, Wiki
CWEB	Dev	6.7k	148k	2	✓	55	Native	Web
	Test	6.8k	149k	2	✓	55	Native	Web
GHTC	Train?	353k edits only		1	✓	0	Native?	Documentation

Table 3
Human-annotated GEC datasets for English. The top half are commonly used to benchmark GEC systems. A question mark (?) indicates unknown or approximated information. CEFR levels: beginner (A1-A2), intermediate (B1-B2), advanced (C1-C2).

2.2.1 Benchmark English Datasets.

FCE. The First Certificate in English (FCE) corpus (Yannakoudakis, Briscoe, and Medlock 2011) is a public subset of the Cambridge Learner Corpus (CLC) (Nicholls 2003) that consists of 1,244 scripts (∼531k words) written by international learners of English as a second language (L2 learners). Each script typically contains two answers to a prompt in the style of a short essay, letter, or description, and each answer has been corrected by a single annotator who has identified and classified each edit according to a framework of 88 error types (Nicholls 2003) (71 unique error types are represented in the FCE). The authors are all intermediate level (B1-B2 level on the Common European Framework of Reference for Languages (CEFR) (Council of Europe 2001)) and the data is split into a standard training, development and test set. The FCE was used as the official dataset of the HOO-2012 shared task (Dale, Anisimoff, and Narroway 2012), one of the official training datasets of the BEA-2019 shared task (Bryant et al. 2019), and has otherwise commonly been used for grammatical error detection (Rei and Yannakoudakis 2016; Bell, Yannakoudakis, and Rei 2019; Yuan et al. 2021). It also contains essay level scores, as well as other limited metadata about the learner, and has been used for automatic essay scoring (AES) (e.g. Ke and Ng (2019)).

NUCLE/CoNLL. The National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier, Ng, and Wu 2013) consists of 1,397 argumentative essays (∼1.16m

words) written by NUS undergraduate students who needed L2 English language support. The essays, which are approximately C1 level, are written on a diverse range of topics including technology, healthcare, and finance, and were each corrected by a single annotator who identified and classified each edit according to a framework of 28 error types. NUCLE was used as the official training corpus of the CoNLL-2013 and CoNLL-2014 shared tasks (Ng et al. 2013, 2014a) as well as one of the official training datasets of the BEA-2019 shared task (Bryant et al. 2019). The CoNLL-2013 and CoNLL-2014 test sets were annotated under similar conditions to NUCLE and respectively consist of 50 essays each (∼30k words) on the topics of i) surveillance technology and population aging, and ii) genetic testing and social media. The CoNLL-2014 test set was also doubly annotated by 2 independent annotators, resulting in 2 sets of official reference annotations; Bryant and Ng (2015) and Sakaguchi et al. (2016) subsequently collected another 8 sets of annotations each for a total of 18 sets of reference annotations. The CoNLL-2013 dataset is now occasionally used as a development set, while the CoNLL-2014 dataset is one of the most commonly used benchmark test sets. One limitation of the CoNLL-2014 test set is that it is not very diverse given that it consists entirely of essays written by a narrow range of learners on only two different topics.

Lang-8. The Lang-8 Corpus of Learner English (Mizumoto et al. 2012; Tajiri, Komachi, and Matsumoto 2012) is a preprocessed subset of the multilingual Lang-8 Learner Corpus (Mizumoto et al. 2011), which consists of 100,000 submissions (∼11.8m words) to the language learning social network service, Lang-8.[2] The texts are wholly unconstrained by topic, and hence include the full range of ability levels (A1-C2), and were written by international L2 English language learners with a bias towards Japanese L1 speakers. Although Lang-8 is one of the largest publicly available corpora, it is also one of the noisiest as corrections are provided by other users rather than professional annotators. A small number of submissions also contain multiple sets of corrections, but all annotations are provided as parallel text and so do not contain explicit edits or error types. Lang-8 was also one of the official training datasets of the BEA-2019 shared task (Bryant et al. 2019).

JFLEG. The Johns Hopkins Fluency-Extended GUG corpus (JFLEG) (Napoles, Sakaguchi, and Tetreault 2017) is a collection of 1,501 sentences (∼28.1k words) split roughly equally into a development and test set. The sentences were randomly sampled from essays written by L2 learners of English of an unspecified ability level (Heilman et al. 2014) and corrected by crowdsourced annotators on Amazon Mechanical Turk (Crowston 2012). Each sentence was annotated a total of 4 times, resulting in 4 sets of parallel reference annotations, but edits were not explicitly defined or classified. The main innovation of JFLEG is that sentences were corrected to be fluent rather than minimally grammatical (Section 2.1). The main criticisms of JFLEG are that it is much smaller than other test sets, the sentences are presented out of context, and it was not corrected by professional annotators (Napoles, Nădejde, and Tetreault 2019).

W&I+LOCNESS. The Write & Improve (W&I) and LOCNESS corpus (Bryant et al. 2019) respectively consist of 3,600 essays (∼755k words) written by international learners of all ability levels (A1-C2) and 100 essays (∼46.2k words) written by native

2 `http://lang-8.com`

British/American English undergraduates. It was released as the official training, development and test corpus of the BEA-2019 shared task and was designed to be more balanced than other corpora such that there are roughly an equal number of sentences at each ability level: Beginner, Intermediate, Advanced, Native. The W&I essays come from submissions to the Write & Improve online essay-writing platform[3] (Yannakoudakis et al. 2018) and the LOCNESS essays, which only comprise part of the development and test sets, come from the LOCNESS corpus (Granger 1998). The training and development set essays were each corrected by a single annotator, while the test set essays were corrected by 5 annotators resulting in 5 sets of parallel reference annotations. Edits were explicitly defined, but not manually classified, so error types were added automatically using the ERRANT framework (Bryant, Felice, and Briscoe 2017). The test set references are not currently publicly available, so all evaluation on this dataset is done via the BEA-2019 Codalab competition platform,[4] which ensures all systems are evaluated in the same conditions.

2.2.2 Other English Datasets.

CLC. The Cambridge Learner Corpus (CLC) (Nicholls 2003) is a proprietary collection of over 130,000 scripts (∼29.1m words) written by international learners of English (130 different first language backgrounds) for different Cambridge exams of all levels (A1-C2) (Yuan, Briscoe, and Felice 2016; Bryant 2019). It is the superset of the public FCE and annotated in the same way.

EFCamDat. The Education First Cambridge Database (EFCamDat) (Geertzen, Alexopoulou, and Korhonen 2013) consists of 1.18m scripts (∼83.5m words) written by international learners of all ability levels (A1-C2) submitted to the English First online school platform. Approximately 66% of the scripts (∼56.8m words) have been annotated with explicit edits that have been classified according to a framework of 25 error types (Huang et al. 2017). Since the annotations were made by teachers for the purposes of giving feedback to students rather than for GEC system development, they are not always complete (too many corrections may dishearten the learner).

WikEd. The Wikipedia Edit Error Corpus (WikEd) (Grundkiewicz and Junczys-Dowmunt 2014) consists of tens of millions of sentences of revision histories from articles on English Wikipedia. The texts are written and edited by native speakers rather than L2 learners and not all changes are grammatical edits; e.g. information updates. A preprocessed version of the corpus is available[5] (28.5m sentences, 626m words) which filters and modifies sentences such that they only contain edits similar to those in NUCLE. The corpus also includes tools to facilitate the collection of similar Wiki-based corpora for other languages.

AESW. The Automatic Evaluation of Scientific Writing (AESW) dataset consists of 316k paragraphs (∼35.5m words) extracted from 9,919 published scientific journal articles and split into a training, development and test set for the AESW shared task

3 https://writeandimprove.com/
4 https://www.cl.cam.ac.uk/research/nl/bea2019st/#instr
5 https://github.com/snukky/wikiedits

(Daudaravicius et al. 2016). A majority of the paragraphs come from Physics, Mathematics and Engineering journals and were written by advanced or native speakers. The articles were edited by professional language editors who explicitly identified the required edits but did not classify them by error type. Although large, one of the main limitations of the AESW dataset is that the texts come from a very specific domain and many sentences contain placeholder tokens for mathematical notation and reference citations which do not generalise to other domains.

GMEG. The Grammarly Multi-domain Evaluation for GEC (GMEG) dataset (Napoles, Nădejde, and Tetreault 2019) consists of 5,919 sentences (∼122.4k words) split approximately equally across 3 different domains: formal native, informal native, and learner text. Specifically, the formal text is sampled from the WikEd corpus (Grundkiewicz and Junczys-Dowmunt 2014), the informal text is sampled from Yahoo Answers, and the learner text is sampled from the FCE (Yannakoudakis, Briscoe, and Medlock 2011). The sentences were sampled at the paragraph level (except for WikEd) to include some context and were annotated by 4 professional annotators to produce 4 sets of alternative references. One of the goals of GMEG was to diversify researchers away from purely L2 learner-based corpora.

CWEB. The Corrected Websites (CWEB) dataset (Flachs et al. 2020) consists of 13.6k sentences (297k words) sampled from random paragraphs on the web in the CommonCrawl dataset.[6] Paragraphs were filtered to reduce noise (e.g. non-English and duplicates) and loosely defined as formal ("sponsored") and informal ("generic") based on the domain of the URL. The paragraphs, which are split equally between a development set and a test set, were doubly annotated by 2 professional annotators and edits were extracted and classified automatically using ERRANT (Bryant, Felice, and Briscoe 2017). Like GMEG, one of the aims of CWEB was to introduce a dataset that extended beyond learner corpora.

GHTC. The GitHub Typo Corpus (GHTC) (Hagiwara and Mita 2020) consists of 353k edits from 203k commits to repositories in the GitHub software hosting website.[7] All the edits were gathered from repositories that met certain conditions (e.g. a permissive license) and from commits that contained the word 'typo' in the commit message. The intuition behind the corpus was that developers often make small commits to correct minor spelling/grammatical errors and that these annotations can be used for GEC. The main limitation of GHTC is that the majority of edits are spelling or orthographic errors from a specific domain (i.e. software documentation) and that the context of the edit is not always a full sentence.

2.3 Non-English Datasets

Although most work on GEC has focused on English, corpora for other languages are also slowly being created and publicly released for the purposes of developing GEC models. This section introduces some of the most prominent (Table 4), along with other relevant resources, but is again by no means an exhaustive list. These resources are ultimately

6 https://commoncrawl.org/
7 https://github.com/

Language	Corpus	Use	Sents	Toks	Refs	Edit Spans	Error Types	Level	Domain
Arabic	QALB-2014	Train	19.4k*	1m	1	✓	7	Native	Web
		Dev	1k*	53.8k	1	✓	7	Native	Web
		Test	948*	51.3k	1	✓	7	Native	Web
	QALB-2015	Train	310*	43.3k	1	✓	7	A1-C2	Essays
		Dev	154*	24.7k	1	✓	7	A1-C2	Essays
		Test	158*	22.8k	1	✓	7	A1-C2	Essays
		Test	920*	48.5k	1	✓	7	Native	Web
Chinese	NLPTEA-2020	Train	1.1k†	36.9k‡	1	✓	4	A1-C2	Exams
		Test	1.4k†	55.2k‡	1	✓	4	A1-C2	Exams
	NLPCC-2018	Train	717k	14.1m‡	1-21	✗	0	A1-C2?	Web
		Test	2k	61.3k‡	1-2	✓	4	A1-C2?	Essays
	MuCGEC	Dev	1.1k	50k‡	2.3	✓	19	A1-C2?	Exams
		Test	5.9k	228k‡	2.3	✓	19	A1-C2?	Essays, Exams, Web
Czech	AKCES-GEC	Train	42.2k	447k	1	✓	25	A1-Native	Essays, Exams
		Dev	2.5k	28.0k	2	✓	25	A1-Native	Essays, Exams
		Test	2.7k	30.4k	2	✓	25	A1-Native	Essays, Exams
	GECCC	Train	66.6k	750k	1	✓	65	A1-Native	Essays, Exams, Web
		Dev	8.5k	101k	1-2	✓	65	A1-Native	Essays, Exams, Web
		Test	7.9k	98.1k	2	✓	65	A1-Native	Essays, Exams, Web
German	Falko-MERLIN	Train	19.2k	305k	1	✓	56	A1-C2	Essays, Exams
		Dev	2.5k	39.5k	1	✓	56	A1-C2	Essays, Exams
		Test	2.3k	36.6k	1	✓	56	A1-C2	Essays, Exams
Japanese	TEC-JL	Test	1.9k	41.5k‡	2	✗	0	A1-C2?	Forum
Russian	RULEC-GEC	Train	5k	83.4k	1	✓	23	C1-C2	Essays
		Dev	2.5k	41.2k	1	✓	23	C1-C2	Essays
		Test	5k	81.7k	1	✓	23	C1-C2	Essays
Ukrainian	UA-GEC	Train	18.2k	285k	1	✓	4	B1-Native	Essays, Fiction
		Test	2.5k	43.5k	1	✓	4	B1-Native	Essays, Fiction

* The Arabic datasets are split into documents rather than sentences.
† The Chinese NLPTEA datasets are split into paragraphs (1-5 sentences) rather than sentences.
‡ The Chinese and Japanese datasets are split into characters rather than tokens.

Table 4
Human-annotated GEC datasets for non-English languages. A question mark (?) indicates unknown or approximated information. CEFR levels: beginner (A1-A2), intermediate (B1-B2), advanced (C1-C2).

helping to pave the way for research into multilingual GEC (Náplava and Straka 2019; Katsumata and Komachi 2020; Rothe et al. 2021).

Arabic. The Qatar Arabic Language Bank (QALB) project (Zaghouani et al. 2014) is an initiative that aims to collect large corpora of annotated Arabic for the purposes of Arabic GEC system development. A subset of this corpus was used as the official training, development and test data of the QALB-2014 and QALB-2015 shared tasks on Arabic text correction (Mohit et al. 2014; Rozovskaya et al. 2015). In particular, QALB-2014 released 21.3k documents (1.1m words) of annotated user comments submitted to the Al Jazeera news website by native speakers, while QALB-2015 released 622 documents (90.8k words) of annotated essays written by the full range of Arabic L2 learners (A1-C2) (Zaghouani et al. 2015) along with an additional 920 documents (48.5k words) of unreleased Al Jazeera comments. QALB-2015 thus had 2 test sets: one on native Al Jazeera data and one on Arabic L2 learner essays. In all cases, files were provided at the document level (rather than the sentence level) and edits were explicitly identified by trained annotators and classified automatically using a framework of 7 error types.

Chinese. The Test of Chinese as a Foreign Language (TOCFL) corpus (Lee, Tseng, and Chang 2018) and the *Hanyu Shuiping Kaoshi* (HSK: Chinese Proficiency Test) corpus[8] (Zhang 2009) respectively consist of 2.8k essays (1m characters) and 11k essays (4m characters) written by the full range of language learners (A1-C2) who took Mandarin Chinese language proficiency exams. Various subsets of these corpora were used as the official training and test sets in the NLPTEA series of shared tasks on Chinese Grammatical Error Diagnosis (i.e. error detection) between 2014-2020 (Yu, Lee, and Chang 2014; Rao, Yang, and Zhang 2020). The most recent of these shared tasks, NLPTEA-2020, released a total of 2.6k paragraphs (92.1k characters, 1-5 sentences each), which were annotated by a single annotator according to a framework of 4 error types: Redundant (R), Missing (M), Word Selection (S) or Word Order (W).

The NLPCC-2018 shared task (Zhao et al. 2018), which was the first shared task on full error correction in Mandarin Chinese, released a further 717k training sentences (14.1m characters) which were extracted from a cleaned subset of Lang-8 user submissions (Mizumoto et al. 2011). Like the Lang-8 Corpus of Learner English, the ability level of the authors in this dataset is unknown and corrections were provided by other users. The test data for this shared task came from the PKU Chinese Learner Corpus and consists of 2000 sentences (61.3k characters) written by foreign college students. All test sentences were first annotated by a single annotator, who also classified edits according to the same 4-error-type framework as NLPTEA, and subsequently checked by a second annotator who was allowed to make changes to the annotations if necessary.

The Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction (MuCGEC) Zhang et al. (2022b) is a new corpus that is intended to be a more robust test set for Chinese GEC. It contains a total of 7063 sentences (∼278k characters) sampled approximately equally from the NLPCC-2018 training set (Lang-8), the NLPCC-2018 test set (PKU Chinese Learner Corpus) and the NLPTEA-2018/2020 test sets (HSK Corpus). All sentences were annotated by multiple annotators, but identical references were removed, so we report an average of 2.3 references per sentence (90% of all sentences have 1-3 references). Edits were also classified according to a scheme of 19 error types, including 5 main error types and 14 minor sub-types.

Czech. The AKCES-GEC corpus (Náplava and Straka 2019) consists of 47.3k sentences (505k words) written by both learners of Czech as a second language (from both Slavic and non-Slavic backgrounds) and Romani children who speak a Czech ethnolect as a first language. The essays and exam-style scripts come from the Learner Corpus of Czech as a Second Language (CzeSL) (Rosen 2016) which falls under the larger Czech Language Acquisition Corpora (AKCES) project (Šebesta 2010). The essays in the training set were annotated once (1 set of annotations) and the essays in the development and test sets were annotated twice (2 sets of annotations), all with explicit edits that were classified according to a framework of 25 error types.

The Grammar Error Correction Corpus for Czech (GECCC) (Náplava et al. 2022) is an extension of AKCES-GEC that includes both formal texts written by native Czech primary and secondary school students as well as informal website discussions on Facebook and Czech news websites, in addition to the texts written by Czech language learners and Romani children. The total corpus consists of 83k sentences (949k words), all of which were manually annotated (or re-annotated in order to preserve annotation style)

8 http://yuyanziyuan.blcu.edu.cn/en/info/1043/1501.htm

by 5 experienced annotators who explicitly identified edits. Edits were then classified automatically by a variant of ERRANT (Bryant, Felice, and Briscoe 2017) for Czech which included a customised tagset of 65 errors types. GECCC is currently one of the largest non-English corpora and is also larger than most popular English benchmarks.

German. The Falko-MERLIN GEC corpus (Boyd 2018) consists of 24k sentences (381k words) written by learners of all ability levels (A1-C2). Approximately half the data comes from the Falko corpus (Reznicek et al. 2012), which consists of minimally-corrected advanced German learner essays (C1-C2), while the other half comes from the MERLIN corpus (Boyd et al. 2014), which consists of standardised German language exam scripts from a wide range of ability levels (A1-C1). Edits were not explicitly annotated, but extracted and classified automatically using a variation of ERRANT (Bryant, Felice, and Briscoe 2017) which was adapted for German and included a customised tagset for German error types.

Japanese. The TMU Evaluation Corpus for Japanese Learners (TEC-JL) (Koyama et al. 2020) consists of 1.9k sentences (41.5k characters) written by language learners of unknown level (A1-C2?) and submitted to the language learning social network service Lang-8. TEC-JL is a subset of the multilingual Lang-8 Learner Corpus (Mizumoto et al. 2011) and was doubly annotated by 3 native Japanese university students (2 sets of annotations) to be a more reliable test set than the original Lang-8 Learner Corpus which can be quite noisy.

Russian. The Russian Learner Corpus of Academic Writing (RULEC) (Alsufieva, Kisselev, and Freels 2012) consists of essays written by L2 university students and heritage Russian speakers in the United States. A subset of this corpus, 12.5k sentences (206k words), was annotated by 2 native speakers of Russian with backgrounds in linguistics and released as the RULEC-GEC corpus (Rozovskaya and Roth 2019). Edits were explicitly annotated and classified according to a framework of 23 error types. Another corpus of annotated Russian errors, the Russian Lang-8 corpus (RU-Lang8) (Trinh and Rozovskaya 2021), which is a subset of the aforementioned multilingual Lang-8 Learner Corpus (Mizumoto et al. 2011), was also recently announced, however the data has not yet been publicly released.

Ukrainian. The UA-GEC corpus (Syvokon and Nahorna 2021) consists of 20.7k sentences (329k words) written by almost 500 authors from a wide variety of backgrounds (mostly technical and humanities) and ability levels (two-thirds native). The texts cover a wide range of topics including short essays (formal, informal, fictional or journalistic) and translated works of literature, and were annotated by two native speakers with degrees in Ukrainian linguistics. Edits were explicitly annotated and classified according to a scheme of 4 error types: Grammar, Spelling, Punctuation or Fluency.

3. Core Approaches

This section introduces some of the core approaches to GEC including classifiers (statistical and neural), machine translation (statistical and neural), edit-based approaches and language models. We provide a high level overview of how each of these approaches works and highlight notable models that have led to breakthroughs in system development. These approaches provide the foundation on which additional techniques (Section 4) and artificial error generation (Section 5) are built.

3.1 Classifiers

Machine learning classifiers were historically one of the most popular approaches to GEC. The main reason for this was that some of the most common error types for English as a second language (ESL) learners, such as article and preposition errors, have small confusion sets and so are well-suited to multi-class classification. For example, it is intuitive to build a classifier that predicts one of {*a/an*, *the*, ϵ} before every noun phrase in a sentence. To do this, a classifier receives a number of features representing the context of the analysed word or phrase in a sentence and outputs a predicted class that constitutes a correction. Errors are flagged and corrected by comparing the original word used in the text with the most likely candidate predicted by the classifier. This approach has been applied to several common error types including:

- articles (Lee 2004; Han, Chodorow, and Leacock 2006; De Felice 2008; Gamon et al. 2008; Gamon 2010; Dahlmeier and Ng 2011b; Kochmar, Andersen, and Briscoe 2012; Rozovskaya and Roth 2013, 2014);
- prepositions (Chodorow, Tetreault, and Han 2007; De Felice 2008; Gamon et al. 2008; Tetreault and Chodorow 2008; Gamon 2010; Dahlmeier and Ng 2011b; Kochmar, Andersen, and Briscoe 2012; Rozovskaya and Roth 2013, 2014);
- noun number (Berend et al. 2013; van den Bosch and Berck 2013; Jia, Wang, and Zhao 2013; Xiang et al. 2013; Yoshimoto et al. 2013; Kunchukuttan, Chaudhury, and Bhattacharyya 2014);
- verb form (Lee and Seneff 2008; Tajiri, Komachi, and Matsumoto 2012; van den Bosch and Berck 2013; Jia, Wang, and Zhao 2013; Rozovskaya and Roth 2013, 2014; Rozovskaya, Roth, and Srikumar 2014).

Training examples consisting of native and/or learner data are represented as vectors of features that are considered useful for the error type. Since the most useful features often depend on the word class, it is necessary to build separate classifiers for each error type and most of the prior classification-based approaches have focused on feature engineering. For the vast majority of syntactically-motivated errors, features such as contextual word and part-of-speech (POS) n-grams, lemmas, phrase constituency information and dependency relations are generally useful (Felice and Yuan 2014b; Leacock et al. 2014; Rozovskaya and Roth 2014; Wang et al. 2021). The details of training vary depending upon the classification algorithm, but popular examples include naive Bayes (Rozovskaya and Roth 2011; Kochmar, Andersen, and Briscoe 2012), maximum entropy (Lee 2004; Han, Chodorow, and Leacock 2006; Chodorow, Tetreault, and Han 2007; De Felice 2008), decision trees (Gamon et al. 2008), support-vector machines (Putra and Szabó 2013), and the averaged perceptron (Rozovskaya and Roth 2010a,b, 2011).

More recently, neural network techniques have been applied to classification-based GEC, where neural classifiers have been built using context words with pre-trained word embeddings, like Word2Vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014). Different neural network models have been proposed, including convolutional neural networks (CNN) (Sun et al. 2015), recurrent neural networks (RNN) (Wang, Li, and Lin 2017; Li et al. 2019), and pointer networks (Li et al. 2019).

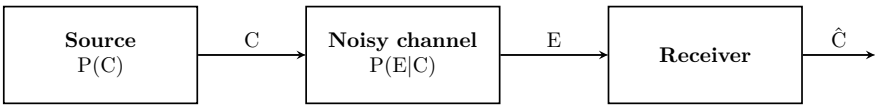


Figure 1
The noisy channel model (Shannon 1948).

One limitation of these classifiers, however, is that they only target very specific error types with small confusion sets and do not extend well to errors involving open-class words (such as word choice errors). Another weakness is that they heavily rely on local context and treat errors independently, assuming that there is only one error in the context and all the surrounding information is correct. When multiple classifiers are combined for multiple error types, classifier order also matters and predictions from individual classifiers may become inconsistent (Yuan 2017). These limitations consequently mean classifiers are generally no longer explored in GEC in favour of other methods.

3.2 Statistical Machine Translation

In contrast with statistical classifiers, one of the main advantages treating GEC as a statistical machine translation (SMT) problem is that SMT can theoretically correct all error types simultaneously without expert knowledge or feature engineering. This also includes interacting errors, which are problematic for rule-based systems and classifiers. Despite originally being developed for translation between different languages, SMT has been successfully applied to GEC, which can be seen as a translation problem from errorful to correct sentences. More specifically, although both the source and target sentences are in the same language, i.e. monolingual translation, the source may contain grammatical errors which should be 'translated' to appropriate corrections. SMT is inspired by the *noisy channel model* (Shannon 1948) and is mathematically formulated using Bayes' rule:

$$\hat{C} = \arg\max_{C} P(C|E) = \arg\max_{C} \frac{P(E|C)P(C)}{P(E)} = \arg\max_{C} P(E|C)P(C) \qquad (1)$$

where a correct sentence C is said to have passed through a noisy channel to produce an erroneous sentence E, and the goal is to reconstruct the correct sentence $\hat{C}$ using a language model (LM) $P(C)$ and a translation model (TM) $P(E|C)$ - see Figure 1. Candidate sentences are generated by means of a decoder, which normally uses a beam search strategy. The denominator $P(E)$ in Equation 1 is ignored since it is constant across all Cs.

The use of SMT for GEC was pioneered by Brockett, Dolan, and Gamon (2006), who built a system to correct errors involving 14 countable and uncountable nouns. Their training data comprised a large corpus of sentences extracted from news articles which were deliberately modified to include artificial mass noun errors. Mizumoto et al. (2011) applied the same techniques to Japanese error correction but improved on them by not only considering a wider set of error types, but also training on real learner examples extracted from the language learning social network website Lang-8. Yuan and Felice (2013) subsequently trained a POS-factored SMT system to correct five types of errors in learner text for the CoNLL-2013 shared task, and revealed the potential of using SMT as a

general approach for correcting multiple error types and interacting errors simultaneously. In the following year, the two top-performing systems in the CoNLL-2014 shared task demonstrated that SMT yielded state-of-the-art performance on general error correction in contrast with other methods (Felice et al. 2014; Junczys-Dowmunt and Grundkiewicz 2014). This success led to SMT becoming a dominant approach in the field and inspired other researchers to adapt SMT technology for GEC, including:

- Adding GEC-specific features to the model to allow for the fact that most words translate into themselves and errors are often similar to their correct forms. Two types of these features include the Levenshtein distance (Felice et al. 2014; Junczys-Dowmunt and Grundkiewicz 2014, 2016; Yuan, Briscoe, and Felice 2016; Grundkiewicz and Junczys-Dowmunt 2018) and edit operations (Junczys-Dowmunt and Grundkiewicz 2016; Chollampatt and Ng 2017; Grundkiewicz and Junczys-Dowmunt 2018).
- Tuning parameter weights with different algorithms, including minimum error rate training (MERT) (Kunchukuttan, Chaudhury, and Bhattacharyya 2014; Junczys-Dowmunt and Grundkiewicz 2014), the margin infused relaxed algorithm (MIRA) (Junczys-Dowmunt and Grundkiewicz 2014), and pairwise ranking optimization (PRO) (Junczys-Dowmunt and Grundkiewicz 2016).
- Training additional large-scale LMs on monolingual native data, such as the British National Corpus (BNC) (Yuan 2017), Wikipedia (Junczys-Dowmunt and Grundkiewicz 2014; Chollampatt and Ng 2017), and Common Crawl (Junczys-Dowmunt and Grundkiewicz 2014, 2016; Chollampatt and Ng 2017).
- Introducing neural network components, such as a neural network global lexicon model (NNGLM) and neural network joint model (NNJM) (Chollampatt, Taghipour, and Ng 2016; Chollampatt and Ng 2017).

Despite their success in GEC, SMT-based approaches suffer from a few shortcomings. In particular, they i) tend to produce locally well-formed phrases with poor overall grammar, ii) exhibit a predilection for changing phrases to more frequent versions even when the original is correct, resulting in unnecessary corrections, iii) are unable to process long-range dependencies and iv) are hard to constrain to particular error types (Felice 2016; Yuan 2017). Last but not least, the performance of SMT systems depends heavily on the amount and quality of parallel data available for training, which is very limited in GEC. A common solution to this problem is to generate artificial datasets, where errors are injected into well-formed text to produce pseudo-incorrect sentences, as described in Section 5.

3.3 Neural Machine Translation

With the advent of deep learning and promising results reported in machine translation and other sequence-to-sequence tasks, neural machine translation (NMT) was naturally extended to GEC. Compared to SMT, NMT uses a single large neural network to model the entire correction process, eliminating the need for complex GEC-specific

feature engineering. Training an NMT system is furthermore an end-to-end process and so does not require separately trained and tuned components as in SMT. Despite its simplicity, NMT has achieved state-of-the-art performance on various GEC tasks (Flachs, Stahlberg, and Kumar 2021; Rothe et al. 2021).

NMT employs the *encoder–decoder* framework (Cho et al. 2014). An encoder first reads and encodes an entire input sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$ into hidden state representations, and a decoder then generates an output sequence $\mathbf{y} = (y_1, y_2, ..., y_{T'})$ by predicting the next word y_t based on the input sequence $\mathbf{x}$ and all the previously generated words $\{y_1, y_2, ..., y_{t-1}\}$:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T'} p(y_t|\{y_1, y_2, ..., y_{t-1}\}, \mathbf{x}) \tag{2}$$

Different network architectures have been proposed for building the encoders and decoders; three commonly used sequence-to-sequence models are RNNs (Bahdanau, Cho, and Bengio 2015), CNNs (Gehring et al. 2017), and Transformers (Vaswani et al. 2017).

3.3.1 Recurrent Neural Networks.

Recurrent Neural Networks (RNN) are a type of neural network that is specifically designed to process sequential data. RNNs are used to transform a variable-length input sequence to another variable-length output sequence (Cho et al. 2014; Sutskever, Vinyals, and Le 2014). To handle long-term dependencies, gated units are usually used in RNNs (Goodfellow, Bengio, and Courville 2016). The two most effective RNN gates are Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Units (GRU) (Cho et al. 2014). Bahdanau, Cho, and Bengio (2015) introduced an attention mechanism to implement variable-length representations, which eased optimisation difficulty and resulted in improved performance. Yuan and Briscoe (2016) presented the first work on NMT-based approach for GEC. Their model consists of a bidirectional RNN encoder and an attention-based RNN decoder. Xie et al. (2016) proposed the use of a character-level RNN sequence-to-sequence model for GEC. Following their work, a hybrid model with nested attention at both the word and character level was later introduced by Ji et al. (2017).

3.3.2 Convolutional Neural Networks.

Another way of processing sequential data is by using a convolutional neural network (CNN) across a temporal sequence. CNNs are a type of neural network that is designed to process grid-like data and specialises in capturing local dependencies (Goodfellow, Bengio, and Courville 2016). CNNs were first applied to NMT by Kalchbrenner and Blunsom (2013), but they were not as successful as RNNs until Gehring et al. (2017) stacked several CNN layers followed by non-linearities. Inspired by this work, Chollampatt and Ng (2018a) proposed a 7-layer CNN sequence-to-sequence model for GEC. In their model, local context is captured by the convolution operations performed over smaller windows and wider context is captured by the multi-layer structure. Their model was the first NMT-based model that significantly outperformed prior SMT-based models. This model was later used in combination with Transformers to build a state-of-the-art GEC system (Yuan et al. 2019).

3.3.3 Transformers.

The Transformer (Vaswani et al. 2017) is the first sequence transducer network that entirely relies on a self-attention mechanism to compute the representations of its input, without the need for recurrence or convolution. Its architecture allows better parallelisation on multiple GPUs, overcoming the weakness of RNNs.

The Transformer has become the architecture of choice for machine translation since its inception (Edunov et al. 2018; Wang et al. 2019; Liu et al. 2020). Previous work has investigated the adaptation of NMT to GEC, such as optimising the model with edit-weighted loss (Junczys-Dowmunt et al. 2018) and adding a copy mechanism (Zhao et al. 2019; Yuan et al. 2019). A copy mechanism allows the model to directly copy tokens from the source sentence, which often has substantial overlap with the target sentence in GEC. The Copy-Augmented Transformer has become a popular alternative architecture for GEC (Hotate, Kaneko, and Komachi 2020; Wan, Wan, and Wang 2020). Another modification to the Transformer architecture is altering the encoder-decoder attention mechanism in the decoder to accept and make use of additional context. For example, Kaneko et al. (2020) added the BERT representation of the input sentence as additional context for GEC, while Yuan and Bryant (2021) added the previous sentences in the document, and Zhang et al. (2022c) added a tree-based syntactic representation of the input sentence.

As the Transformer architecture has a large number of parameters, yet parallel GEC training data is limited, pre-training has become a standard procedure in building GEC systems. The first Transformer-based GEC system (Junczys-Dowmunt et al. 2018) pre-trained the Transformer decoder on a language modeling task, but it has since become more common to pre-train on synthetic GEC data. The top two systems in the BEA-2019 shared task (Grundkiewicz, Junczys-Dowmunt, and Heafield 2019; Choe et al. 2019) and a recent state-of-the-art GEC system (Stahlberg and Kumar 2021) both pre-trained their Transformer models with synthetic data, but they generated their synthetic data in different ways. We discuss different techniques for generating synthetic data in Section 5.1. More recently, with the advances in large pre-trained language models, directly fine-tuning large pre-trained language models with GEC parallel data has been shown to achieve comparable performance with synthetic data pre-training (Katsumata and Komachi 2020), even reaching state-of-the-art performance (Rothe et al. 2021; Tarnavskyi, Chernodub, and Omelianchuk 2022).

Irrespective of the type of NMT architecture however (RNN, CNN, Transformer), NMT systems share several weaknesses with SMT systems, most notably in terms of data requirements. In particular, although NMT systems are more capable at correcting longer range and more complex errors than SMT, they also require as much training data as possible, which can lead to extreme resource and time requirements: it is not uncommon for some models to require several days of training time on a cluster of GPUs. Moreover, neural models are almost completely uninterpretable (which furthermore makes them difficult to customise) and it is nearly impossible for a human to determine the reasoning behind a given decision; this is particularly problematic if we also want to explain the cause of an error to a user rather than just correct it. Ultimately however, a key strength of NMT is that it is an end-to-end approach, and so does not require feature engineering or much human intervention, and it is undeniable that it produces some of the most convincing output to date.

3.4 Edit-based approaches

While most GEC approaches generate a corrected sentence from an input sentence, the edit generation approach generates a sequence of edits to be applied to the input sentence instead. As GEC has a high degree of token copying from the input to the output, Stahlberg and Kumar (2020) argued that generating the full sequence is wasteful. By generating edit operations instead of all tokens in a sentence, the edit generation approach typically has a faster inference speed, reported to be five to ten times faster than GEC systems that generate the whole sentence. One limitation of this approach, however, is that edit operations tend to be token-based, and so sometimes fail to capture more complex, multi-token fluency edits (Lai et al. 2022). Edit generation has been cast as a sequence tagging task (Malmi et al. 2019; Awasthi et al. 2019; Omelianchuk et al. 2020; Tarnavskyi, Chernodub, and Omelianchuk 2022) or a sequence-to-sequence task (Stahlberg and Kumar 2020).

In the sequence tagging approach, for each token of an input sentence, the system predicts an edit operation to be applied to that token (Table 5). This approach requires the user to define a set of tags representing the edit operations to be modelled by the system. Some edits can be universally modelled, such as conversion of verb forms or conversion of nouns from singular to plural form. Some others such as word insertion and word replacement are token-dependent. Token-dependent edits need a different tag for each possible word in the vocabulary, resulting in the number of tags growing linearly with the number of unique words in the training data. Thus, the number of token-dependent tags to be modelled in the system becomes a trade-off between coverage and model size.

Source	After	many	years	he	still	dream	to	become	a	super	hero
Target	After	many	years	, he	still	dreams	of	becoming	a	super	hero
Edits	KEEP	KEEP	APP_,	KEEP	KEEP	VB_VBZ	REP_of	VB_VBG	KEEP	KEEP	KEEP

Table 5
Example task formulation of edit generation in the sequence tagging approach from (Omelianchuk et al. 2020). APP_x denotes an operation of appending token x, and REP_x denotes replacing the current token with x.

On the other hand, the sequence-to-sequence approach is more flexible as it does not limit the output to pre-defined edit operation tags. It produces a sequence of edits, each consisting of a span position, a replacement string, and an optional tag for edit type (Table 6). These tags add interpretability to the process and have been shown to improve model performance. As generation in the sequence-to-sequence approach has a left-to-right dependency, the inference procedure is slower than that in the sequence tagging approach. It is still five times faster than that in the whole sentence generation approach as the edit sequence generated is much shorter than the sequence of all tokens in the sentence (Stahlberg and Kumar 2020).

The main advantages of edit-based approaches to GEC are thus that they not only add much needed transparency and explainability to the correction process, but they are also much faster at inference time than NMT. Their main disadvantages, however, are that they generally require human engineering to define the size and scope of the edit label set, and that it is more difficult to represent interacting and complex multi-token edits with token-based labels. Like all neural approaches, they also depend on as much training data as possible, but when data is available, edit-based approaches are very competitive with state-of-the-art NMT models.

Source	After many years he still dream to become a super hero .
Target	After many years , he still dreams of becoming a super hero .
Edits	(SELF,3,SELF), (PUNCT,3,‘,’), (SELF,5,SELF), (SVA,6,‘dreams’), (PART,7,‘of’), (FORM,8,‘becoming’), (SELF,12,SELF)

Table 6
Example task formulation of edit generation in the sequence-to-sequence approach from (Stahlberg and Kumar 2020). Each tuple represents a tag, a span's ending position, and a replacement string.

3.5 Language Models for Low-Resource and Unsupervised GEC

Unlike previous strategies, language model based GEC does not require training a system with parallel data. Instead, it employs various techniques using n-gram or Transformer language models. LM-based GEC was a common approach before machine translation-based GEC became popular (Dahlmeier and Ng 2012a; Lee and Lee 2014), but has experienced a recent resurgence with low-resource GEC and unsupervised GEC due to the effectiveness of large Transformer-based language models (Alikaniotis and Raheja 2019; Grundkiewicz and Junczys-Dowmunt 2019; Flachs, Lacroix, and Søgaard 2019). Recent advances have enabled Transformer-based language models to more adequately capture syntactic phenomena (Jawahar, Sagot, and Seddah 2019; Wei et al. 2021), making them capable GEC systems when little or no data is available. These systems can, however, become even more capable when exposed to a small amount of parallel data (Mita and Yanaka 2021).

3.5.1 Language models as Discriminators.

The traditional LM-based approach to GEC makes the assumption that low probability sentences are more likely to contain grammatical errors than high probability sentences, and so a GEC system must determine how to transform the former into the latter based on language model probabilities (Bryant and Briscoe 2018). Correction candidates can be generated from confusion sets (Dahlmeier and Ng 2011a; Bryant and Briscoe 2018), classification-based GEC models (Dahlmeier and Ng 2012a), or finite-state transducers (Stahlberg, Bryant, and Byrne 2019).

Yasunaga, Leskovec, and Liang (2021) proposed an alternative method using the break-it-fix-it (BIFI) approach (Yasunaga and Liang 2021), with a language model as the critic (LM-critic). Specifically, BIFI trains a breaker (noising channel) and a fixer (GEC model) on multiple rounds of feedback loops. An initial fixer is used to correct erroneous text, then the sentence pairs are filtered using LM-critic. Using this filtered data, the breaker is trained and used to generate new synthetic data from a clean corpus. These new sentence pairs are then also filtered using LM-critic and subsequently used to train the fixer again. The BIFI approach can be used for unsupervised GEC by training the fixer on synthetic data.

3.5.2 Language models as Generators.

A more recent LM-based approach to GEC is to use a language model as a zero-shot or few-shot generator to generate a correction given a prompt and noisy input sentence. For example, given the prompt “Correct the grammatical errors in the following

text:" followed by an input sentence, the language model is expected to generate a corrected form of the input sentence given the prompt as context. This approach has become possible largely due to the advent of Large Language Models (LLMs), such as GPT-2 (Radford et al. 2019), GPT-3 (Brown et al. 2020), OPT (Zhang et al. 2022a) and PaLM (Chowdhery et al. 2022), which have been trained on up to a trillion words and parameterised using tens or hundreds of billions of parameters. These models have furthermore been shown to be capable of generalising to new unseen tasks or languages by being fine-tuned on a wide variety of other NLP tasks (Sanh et al. 2022; Wei et al. 2022; Muennighoff et al. 2022), and so it is possible, for the first time, to build a system that is capable of carrying out multilingual GEC without having been explicitly trained to do so. Despite their potential however, we are unaware of any published studies at this time that have formally benchmarked generative LLMs against any of the standard GEC test sets.

Regardless of the type of language model, the main advantage of language model based approaches is that they only require unannotated monolingual data and so are much easier to extend to other languages than all other approaches. While discriminative LMs may not perform as well as state-of-the-art models and generative LLMs models have not been formally benchmarked, LMs have nevertheless proven themselves capable and can theoretically correct all types of errors, including complex fluency errors. The main disadvantage of language model approaches, however, is that it can be hard to adequately constrain the model, and so models sometimes replace grammatical words with other words that simply occur more frequently in a given context. An additional challenge in generative LLM-based GEC is that prompt engineering is important (Liu et al. 2023) and output may vary depending on whether a system was asked to 'correct' a grammatical error or 'fix' a grammatical error. Ultimately, all LM-based approaches suffer from the limitation that probability is not grammaticality, and so rare words may be mistaken for errors.

4. Additional Techniques

While Section 3 introduced the core technologies underpinning modern GEC systems, a number of other techniques are also commonly applied to further boost performance. Several of these techniques are introduced in this section, including re-ranking, ensembling and system combination, multi-task learning, custom inference methods (e.g., iterative decoding), contextual GEC, and Generative Adversarial Networks (GANs).

4.1 Re-ranking

Machine translation based (both SMT and NMT) systems can produce an n-best list of alternative corrections for a single sentence. This has led to much work on n-best list re-ranking, which aims to determine whether the best correction for a sentence is not the most likely candidate produced by the system (i.e. $n = 1$), but is rather somewhere further down the top n most likely candidates (Yuan, Briscoe, and Felice 2016; Mizumoto and Matsumoto 2016; Hoang, Chollampatt, and Ng 2016). As a separate post-processing step, candidates produced by an SMT-based or NMT-based GEC system can be re-ranked using a rich set of features that have not been explored by the decoder before, so that better candidates can be selected as 'optimal' corrections. During re-ranking, GEC-specific features can then be easily adapted without worrying about fine-grained model smoothing issues. In addition to the original model scores of the candidates, useful features include:

- sentence fluency scores calculated from: LMs (Yuan, Briscoe, and Felice 2016; Chollampatt and Ng 2018a), neural error detection models (Yannakoudakis et al. 2017; Yuan et al. 2019), neural quality estimation models (Chollampatt and Ng 2018b), and BERT (Kaneko et al. 2019);
- similarity measures like Levenshtein Distance (Yannakoudakis et al. 2017; Yuan et al. 2019) and edit operations (Chollampatt and Ng 2018a; Kaneko et al. 2019);
- length-based features (Yuan, Briscoe, and Felice 2016);
- right-to-left models (Grundkiewicz, Junczys-Dowmunt, and Heafield 2019; Kaneko et al. 2020);
- syntactic features like POS n-grams, dependency relations (Mizumoto and Matsumoto 2016);
- error detection information which has been used in a binary setting (Yannakoudakis et al. 2017; Yuan et al. 2019), as well as a multi-class setting (Yuan et al. 2021).

N-best list reranking has traditionally been one of the simplest and most popular methods of boosting system performance. An alternative form of reranking is to collect all the edits from the N-best corrections and filter them using an edit-scorer (Sorokin 2022).

4.2 Ensembling and System Combination

Ensembling is a common technique in machine learning to combine the predictions of multiple individually trained models. Ensembles often generate better predictions than any of the single models that are combined (Opitz and Maclin 1999). In GEC, ensembling usually refers to averaging the probabilities of individually trained GEC models when predicting the next token in the sequence-to-sequence approach or the edit tag in the edit-based approach. GEC models that are combined into ensembles usually have similar properties with only slight variations, which can be the random seed (Stahlberg and Kumar 2021), the pre-trained model (Omelianchuk et al. 2020), or the architecture (Choe et al. 2019).

On the other hand, different GEC approaches have different strengths and weaknesses. Susanto, Phandi, and Ng (2014) have shown that combining different GEC systems can produce a better system with higher accuracy. When combining systems that have substantial differences, training a system combination model is preferred over ensembles. A system combination model allows the combined system to properly integrate the strengths of the GEC systems and has been shown to produce better results than ensembles (Kantor et al. 2019; Qorib, Na, and Ng 2022). The combination model can be trained through learning the characteristic of the GEC systems (Kantor et al. 2019; Lin and Ng 2021; Qorib, Na, and Ng 2022) or learning how to score a correction by supplying the model with examples of good and bad corrections for different kinds of student sentences (Sorokin 2022). Moreover, most system combination methods for GEC work on a *black-box* setup (Kantor et al. 2019; Lin and Ng 2021; Qorib, Na, and Ng 2022), only requiring the systems' outputs without any access to the systems' internals and the prediction probabilities. When the individual component systems are not different

enough, encouraging the individual systems to be more diverse before combining them can also improve performance (Han and Ng 2021).

4.3 Multi-task learning

Multi-task learning allows systems to use information from related tasks and learn from multiple objectives via shared representations, leading to performance gains on individual tasks. Rei and Yannakoudakis (2017) was the first to investigate the use of different auxiliary objectives for the task of error detection in learner writing through a neural sequence-labelling model. In addition to predicting the binary error labels (i.e. correct or incorrect), they experimented with also predicting specific error type information, including the learner's L1, token frequency, POS tags and dependency relations. Asano et al. (2019) employed a similar approach in which their error correction model additionally estimated the learner's language proficiency level and performed sentence-level error detection simultaneously. Token-level and sentence-level error detection have also both been explored as auxiliary objectives in NMT-based GEC (Yuan et al. 2019; Zhao et al. 2019), where systems have been trained to jointly generate a correction and predict whether the source sentence (or any token in it) is correct or incorrect. Labels for these auxiliary error detection tasks can be extracted automatically from existing datasets using automatic alignment tools like ERRANT (Bryant, Felice, and Briscoe 2017).

4.4 Custom inference methods

Various inference techniques have been proposed to improve the quality of system output or speed up inference time in GEC. The most common of these, which specifically improves output quality, is to apply multiple rounds of inference, known as *iterative decoding* or *multi-turn decoding*. Since the input and output of GEC are in the same language, the output of the model can be passed through the model again to produce a second iteration of output. The advantage of this is that the model gets a second chance to correct errors it might have missed during the first iteration. Lichtarge et al. (2019) thus proposed an iterative decoding algorithm that allows a model to make multiple incremental corrections. In each iteration, the model is allowed to generate a different output only if it has high confidence. This technique proved effective for GEC systems trained on noisy data such as Wikipedia edits, but not as effective on GEC systems trained on clean data. Ge, Wei, and Zhou (2018) proposed an alternative iterative decoding technique called fluency boost, in which the model performs multiple rounds of inference until a fluency score stops increasing, while Lai et al. (2022) proposed an iterative approach that investigated the effect of correcting different types of errors (missing, replacement, unnecessary words) in different orders. Iterative decoding is commonly employed in sequence-labelling GEC systems which cannot typically correct all errors in a single pass. In these systems, iterative decoding is applied until the model stops making changes to the output or the number of iterations reaches a limit (Awasthi et al. 2019; Omelianchuk et al. 2020; Tarnavskyi, Chernodub, and Omelianchuk 2022).

Other inference techniques have been proposed to speed up inference time in GEC. As many tokens in GEC are copied from the input to the output, standard left-to-right inference can be inefficient. Chen et al. (2020a) thus proposed a two-step process that only performs correction on text spans that are predicted to contain grammatical errors. Specifically, their system first predicts erroneous spans using an erroneous span detection (ESD) model, and then corrects only the detected spans using an erroneous span correction (ESC) model. They reported reductions in inference time of almost

50% compared to a standard sequence-to-sequence model. In contrast, Sun et al. (2021) proposed a parallelisation technique to speed up inference, aggressive decoding, which can be applied to any sequence-to-sequence model. Specifically, aggressive decoding first decodes as many tokens as possible in parallel and then only re-decodes tokens one-by-one at the point where the input and predictions differ (if any). When the input and predicted tokens start to match again, aggressive decoding again decodes the remainder in parallel until either the tokens no longer match or the end-of-sentence token is predicted. Since the input and output sequences in GEC are often very similar, this means most tokens can be decoded aggressively, yielding an almost ten time speedup in inference time.

4.5 Contextual GEC

Context provides valuable information that is crucial for correcting many types of grammatical errors and resolving inconsistencies. Existing GEC systems typically perform correction at the sentence-level however, i.e. each sentence is processed independently, and so cross-sentence information is ignored. These systems thus frequently fail to correct contextual errors, such as verb tense, pronoun, run-on sentence and discourse errors, which typically rely on information outside the scope of a single sentence. Corrections proposed by such narrow systems are furthermore likely to be inconsistent throughout a paragraph or entire document.

Chollampatt, Wang, and Ng (2019) were the first to address this problem by adapting a CNN sequence-to-sequence model to be more context-aware. Specifically, they introduced an auxiliary encoder to encode the two previous sentences along with the input sentence and incorporated the encoding in the decoder via attention and gating mechanisms. Yuan and Bryant (2021) subsequently compared different architectures for capturing wider context in Transformer-based GEC and showed that local context is useful (≤ 2 sentences) but very long context (> 2 sentences) is not necessary for improved performance.

Since human reference edits are not annotated for whether an error depends on local context or long range context, it is often difficult to evaluate the extent to which a context-aware system improves the correction of context-sensitive errors. Chollampatt, Wang, and Ng (2019) thus constructed a synthetic dataset of verb tense errors which required cross-sentence context for correction, and Yuan and Bryant (2021) proposed a document-level evaluation framework to address this problem.

4.6 Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) are an approach to model training that makes use of both a generator, to generate some output; and a discriminator, to discriminate between real data and artificial output. In the context of GEC, Raheja and Alikaniotis (2020) were the first to apply this methodology to error correction, in which they trained a standard sequence-to-sequence Transformer model to generate grammatical sentences from parallel data (the generator) and a sentence classification model to discriminate between these generated output sentences and human-annotated reference sentences (the discriminator). During training, the models competed adversarially such that the generator learnt to generate corrected sentences that are indistinguishable from the reference sentences (and thus fooled the discriminator), while the discriminator learnt to identify the differences between real and generated sentences

(and thus defeated the generator). This adversarial training process was ultimately shown to produce a better sequence-to-sequence model.

In addition to sequence-to-sequence generation, GANs have also been applied to sequence-labelling for GEC. In particular, Parnow, Li, and Zhao (2021) trained a generator to generate increasingly realistic errors (in the form of token-based edit labels) and a discriminator to differentiate between artificially-generated edits and real human edits. They similarly reported improvements over a baseline that was not trained adversarially.

5. Data Augmentation

A common problem in GEC is that the largest publicly-available high-quality parallel corpora only contain roughly 50k sentence pairs, and larger corpora, such as Lang-8, are noisy (Mita et al. 2020; Rothe et al. 2021). This data sparsity problem has motivated a lot of research into synthetic data generation, especially in the context of resource-heavy NMT approaches, because synthetic data primarily requires a native monolingual source corpus rather than a labour-intensive manual annotation process. In this section, we introduce several different types of data augmentation methods, including rule-based noise injection and back-translation, but also noise reduction which aims to improve the quality of existing datasets by removing/down-weighting noisy examples. It is an open question as to how to best evaluate the quality of synthetic data (Htut and Tetreault 2019; White and Rozovskaya 2020). An effort has been made by (Kiyono et al. 2019) to compare the noise injection method and back-translation, but it is hard to comprehensively compare synthetic data generation methods directly, so most research evaluates it indirectly in terms of its impact on the performance of previous experiments. Data augmentation has nevertheless contributed greatly to GEC system improvement and has become a staple component of recent models.

5.1 Synthetic Data Generation

GEC is sometimes regarded as a low-resource machine translation task (Junczys-Dowmunt et al. 2018). With the dominance of neural network approaches, the need for more data grows as model size continues to increase. However, obtaining human annotations is expensive and difficult. Thus, techniques to generate synthetic parallel corpora from clean monolingual corpora have been intensely explored. A synthetic parallel corpus is generated by adding noises to a sentence and pairing it with the original sentence. The corrupted sentence is then regarded as a learner's sentence (source) and the original clean sentence is regarded as the reference (target). There are many ways to generate synthetic sentences, and the dominant techniques usually fall under the category of noise injection or back-translation (Kiyono et al. 2019).

5.1.1 Noise Injection.

One way to artificially generate grammatical errors to clean monolingual corpora is by perturbing a clean text to make it grammatically incorrect. The perturbations can be in the form of rule-based noising operations or error patterns that usually appear in GEC parallel corpora.

Rule-based. The most intuitive way of adding noise to a clean corpus is by applying a series of perturbation operations based on some pre-defined rules. The rules are applied based

on a probability, which can be decided arbitrarily, empirically, or through some observations of available data. Ehsan and Faili (2013) apply one error to each sentence from predefined error templates that include omitting prepositions, repeating words, and so on. Lichtarge et al. (2019) introduce spelling errors to Wikipedia edit history by performing deletion, insertion, replacement, and transposition of characters. Zhao et al. (2019) also apply a similar noising strategy but at the word level, that is deleting, adding, shuffling, and replacing words in a sentence. Grundkiewicz, Junczys-Dowmunt, and Heafield (2019) combine both approaches, character-level and word-level noising, but word substitution is limited to pairs from a confusion set made from an inverted spellchecker. Similarly, Xu et al. (2019) also combine both approaches but with a more complex word substitution strategy by making use of part-of-speech (POS) tags. The rule-based injection technique can also be applied dynamically during training to increase the error rate in a parallel corpus instead of creating additional training data (Zhao and Wang 2020).

Error patterns. Another way of generating synthetic data is through injecting errors that frequently occur in GEC parallel corpora. In this way, the errors are more similar to the ones that humans usually make. Rozovskaya and Roth (2010b) proposed three different methods of injecting article errors, based on the error distribution in English as a Second Language (ESL) data. They proposed adding article errors based on the distribution of articles in a text before correction, the distribution of articles in the corrected text, and the distribution of article corrections themselves. Felice and Yuan (2014a) later improved the method by taking into consideration the morphology, POS tag, semantic concept, and word sense information of a text when generating the artificial errors. Rei et al. (2017) further extended it to all types of errors. Another direction of emulating human errors is by extracting the correction patterns from GEC parallel corpora and applying the inverse of those corrections on grammatically correct sentences, as done by Yuan and Felice (2013) using the corrections from the NUCLE corpus and by Choe et al. (2019) using the corrections from the W&I training data. The correction patterns are extracted both in lexical form (an → the) and POS (NN → NNS).

5.1.2 Back-translation.

Emulating human errors can be made in a more automated and dynamic way via a noisy channel model. The noisy channel model is trained with the inverse of a GEC parallel corpus, treating the learner's sentence as the target and the reference sentence as the source. This technique is commonly called back-translation. The technique was originally proposed for generating additional data in machine translation (Sennrich, Haddow, and Birch 2016), but it is also directly applicable to GEC. Rei et al. (2017) were the first to apply back-translation to grammatical error detection (GED) and Xie et al. (2018) were the first to apply it to GEC. Yuan et al. (2019) add a form of quality control to Rei et al. (2017) based on language model probabilities in an effort to make sure that the generated synthetic sentences are less probable (and hence hopefully less grammatical) than the original input sentences. Between the rule-based and back-translation strategy, Kiyono et al. (2019) report that the back-translation strategy has better empirical performance. They also compare back-translation with a noisy beam-search strategy (Xie et al. 2018) and back-translation with sampling strategy (Edunov et al. 2018), and report that both achieve competitive performance. Koyama et al. (2021) furthermore compare the effect of using different architectures (e.g. CNN, LSTM, Transformer) for back-translation, and find that interpolating multiple generation systems tends to produce better synthetic data to train a GEC system on.

Another variant of back-translation was proposed by Stahlberg and Kumar (2021) to generate more complex edits. They found that generating a sequence of edits using Seq2Edit (Stahlberg and Kumar 2020) works better than generating the corrupted sentences directly. They also reported that back-translation with sampling worked better than beam search in their experiments.

5.1.3 Round-trip Translation.

A less popular alternative to back-translation is round-trip translation, which generates synthetic sentence pairs via a bridge language; e.g. English-Chinese-English. The assumption is that the MT system will make translation errors and so the output via the bridge language will be noisy in relation to the input. This strategy was employed by Madnani, Tetreault, and Chodorow (2012) and Lichtarge et al. (2019), who furthermore both explored the effect of using different bridge languages. Zhou et al. (2020) explore a similar technique, except use a bridge language as the input to both a low-quality and high-quality translation system (namely SMT vs. NMT), and treat the output from the former as an ungrammatical noisy sentence and the output from the latter as the reference.

5.2 Augmenting Official Datasets

Besides generating synthetic data to address the data sparsity problem in GEC, other works focus on augmenting official datasets, via noise reduction or model enhancement.

Noise reduction aims to reduce the impact of wrong corrections in the official GEC datasets. One direction focuses on correcting noisy sentences. Mita et al. (2020) and Rothe et al. (2021) achieve this by incorporating a well-trained GEC model to reduce wrong corrections. The other direction attempts to down-weight noisy sentences. Lichtarge, Alberti, and Kumar (2020) introduce an offline re-weighting method to score each training sentence based on delta-log perplexity, Δppl, which measures the model's log perplexity difference between checkpoints for a single sentence. Sentences with lower Δppl are preferred and assigned a higher weight during training.

Model enhancement augments official datasets to address the model's weakness. Parnow, Li, and Zhao (2021) aim to enhance performance by reducing the error density mismatch between training and inference. They use a generative adversarial network (GAN) (Goodfellow et al. 2014) to produce an ungrammatical sentence that could better represent the error density at inference time. Lai et al. (2022) also address the mismatch between training and inference, but specific to multi-round inference. They propose additional training stages that make the model consider edit type interdependence when predicting the corrections. Cao, Yang, and Ng (2021) aim to enhance model performance in low-error density domains. The augmented sentences are generated by beam search to capture wrong corrections that the model tends to make. Supervised contrastive learning (Chen et al. 2020b) is then applied to enhance model performance.

6. Evaluation

A core component of any NLP system is the ability to measure model performance. This section hence first introduces the most commonly-used evaluation metrics in GEC, namely the MaxMatch (M^2) scorer (Dahlmeier and Ng 2012b), ERRANT (Bryant, Felice, and Briscoe 2017; Felice, Bryant, and Briscoe 2016) and GLEU

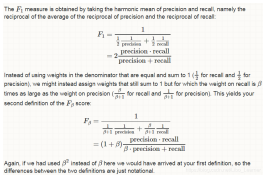
(Napoles et al. 2015, 2016), as well as other reference-based and reference-less metrics that have been proposed. It next discusses the problem of metric reliability, particularly in relation to correlation with human judgements, and explains why it is difficult to draw any robust conclusions. The section concludes with a discussion of best practices in GEC evaluation, including defining standard experimental settings and highlighting their limitations. To date, almost all evaluation in GEC has been carried out at the sentence level.

6.1 MaxMatch

One of the most prevalent evaluation methods used in current GEC research is the MaxMatch (M^2) scorer[9] (Dahlmeier and Ng 2012b) which calculates an F_β-score (van Rijsbergen 1979). Specifically, the M^2 scorer is a reference-based metric which compares system hypothesis edits against human-annotated reference edits and counts a True Positive (TP) if a hypothesis edit matches a reference edit, a False Positive (FP) if a hypothesis edit does *not* match any reference edit, and a False Negative (FN) if a reference edit does *not* match any hypothesis edit. An example of each case is shown below.

		TP		**FN**	**FP**		
Original	I	likes	to	drive	a	bicycle	.
Hypothesis	I	like	to	drive	the	bicycle	.
Reference	I	like	to	ride	a	bicycle	.

The total number of TPs, FPs and FNs for a dataset can then be used to calculate Precision (P) (Equation 3) and Recall (R) (Equation 4), which respectively denote the proportion of hypothesis edits that were correct and the proportion of reference edits that were found in the hypothesis edits, which in turn can be used to calculate the F_β-score (Equation 5). In current GEC research, it is common practice to use $\beta = 0.5$, first introduced in (Ng et al. 2014b), which weights precision twice as much as recall, because it is generally considered more important for a GEC system to be precise than to necessarily correct all errors.



$$(3) \quad P = \frac{TP}{TP + FP} \qquad (4) \quad R = \frac{TP}{TP + FN} \qquad (5) \quad F_\beta = (1 + \beta^2) \times \frac{P \times R}{(\beta^2 \times P) + R}$$

One issue of using edit overlap to measure performance is that there is often more than one way to define an edit. For example, the edit [*has eating* → *was eaten*] can also be realised as [*has* → *was*] and [*eating* → *eaten*]. If the hypothesis combines them, but the reference does not, the edit will not be counted as a TP even though it produces the same valid correction. As a result, system performance is not measured correctly.

The innovation of the M^2 scorer is that it uses a Levenshtein alignment (Levenshtein 1966) between the original text and a system hypothesis to dynamically explore the different ways of combining edits such that the hypothesis edits *maximally match* the reference edits. As such, it overcomes a limitation of the previous scorer used in the HOO shared tasks which could return erroneous scores. Whenever there is more than one set of reference edits for a test sentence, the M^2 scorer tries each set in turn and chooses the one that leads to the best performance for that test sentence.

9 https://www.comp.nus.edu.sg/~nlp/conll14st.html

6.2 ERRANT

The ERRANT scorer[10] (Bryant, Felice, and Briscoe 2017) is similar to the M^2 scorer, in that it is a reference-based metric that measures performance in terms of an edit-based F-score, but differs primarily in that it is also able to calculate error types scores. Specifically, unlike the M^2 scorer, it uses a linguistically-enhanced Damerau-Levenshtein alignment algorithm to extract edits from the hypothesis text (Felice, Bryant, and Briscoe 2016), and then classifies them according to a rule-based error type framework. This facilitates the calculation of F-scores for each error type rather than just overall, which can be invaluable for a detailed system analysis. For example, System A might outperform System B overall, but system B might outperform System A on certain error types, and this information can be used to improve System A.

ERRANT was the first scorer to be able to evaluate GEC systems in terms of error types and is moreover able to do so at three different levels of granularity:

- Edit Operation (3 labels): Missing, Replacement, Unnecessary
- Main Type (25 labels): e.g. Noun, Spelling, Verb Tense
- Full Type (55 labels): e.g. Missing Noun, Replacement Noun, Unnecessary Noun

It is also able to carry out this analysis in terms of both error detection and correction. ERRANT currently only supports English, but other researchers have independently extended it for German (Boyd 2018), Greek (Korre, Chatzipanagiotou, and Pavlopoulos 2021), Arabic (Belkebir and Habash 2021) and Czech (Náplava et al. 2022).

6.3 GLEU

Like M^2 and ERRANT, GLEU[11] (Napoles et al. 2015, 2016) is also a reference-based metric except it does not require explicit edit annotations but rather only corrected reference sentences. It was inspired by the BLEU score (Papineni et al. 2002) commonly used in machine translation and was motivated by the fact that human-annotated edit spans are somewhat arbitrary and time-consuming to collect. The main intuition behind GLEU is that it rewards hypothesis n-grams that overlap with the reference but not the original text, and penalises hypothesis n-grams that overlap with the original text but not the reference. It is important to be aware that GLEU is often attributed to Napoles et al. (2015), but actually implemented according to Napoles et al. (2016), which is a revised formulation. The revised formulation is calculated as follows.

Consider a corpus of original sentences $O = \{o_1, ..., o_k\}$ and their corresponding hypothesis sentences $H = \{h_1, ..., h_k\}$ and reference sentences $R = \{r_1, ..., r_k\}$. For each original, hypothesis and reference sentence, let o_i, h_i and r_i respectively denote the sequences of n-grams of length $n = \{1, 2, ..., N\}$ ($N = 4$ by default in GLEU) in the sentences rather than the sentences themselves. This can then be used to calculate a precision term p_n (Equation 6) that takes the intuition about rewarding or penalising n-gram overlap into account.

10 https://github.com/chrisjbryant/errant
11 `https://github.com/cnap/gec-ranking`

$$p_n = \frac{\sum_{i=1}^{|H|}\left(\sum_{g\in\{h_i\cap r_i\}}\text{count}_{h_i,r_i}(g) - \sum_{g\in\{h_i\cap o_i\}}\max[0,\text{count}_{h_i,o_i}(g)-\text{count}_{h_i,r_i}(g)]\right)}{\sum_{i=1}^{|H|}\sum_{g\in\{h_i\}}\text{count}_{h_i}(g)}$$

$$\begin{aligned}\text{count}_a(g) &= \#\text{ occurrences of n-gram } g \text{ in } a\\ \text{count}_{a,b}(g) &= \min(\#\text{ occurrences of n-gram } g \text{ in } a, \#\text{ occurrences of n-gram } g \text{ in } b)\end{aligned} \tag{6}$$

$$\text{BP} = \begin{cases}1 & \text{if } l_h > l_r\\ \exp(1-l_r/l_h) & \text{if } l_h \leq l_r\end{cases} \tag{7}$$

$$\text{GLEU}(O,H,R) = \text{BP}\cdot\exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n\right) \tag{8}$$

Like the BLEU score, GLEU also has a Brevity Penalty (BP) to penalise hypotheses that are shorter than the references (Equation 7), where l_h denotes the total number of tokens in the hypothesis corpus and l_r denotes the total number of tokens in the *sampled* reference corpus. It is important to note that when there is more than one reference sentence, GLEU iteratively selects one at random and averages the score over 500 iterations. GLEU is finally calculated as in Equation 8.

6.4 Other Metrics

In addition to M^2, ERRANT and GLEU, other metrics have also been proposed in GEC. Some of these are *reference-based*, i.e. they require human-annotated target sentences, while others are *reference-less*, i.e. they do *not* require human-annotated target sentences. This section briefly introduces metrics of both types.

6.4.1 Reference-based Metrics.

I-measure. The *I*-measure (Felice and Briscoe 2015) was designed to overcome certain shortcomings of the M^2 scorer, e.g. the M^2 scorer is unable to differentiate between a bad system (TP=0, FP>0) and a do-nothing system (TP=0, FP=0) which both result in F=0, and instead measure system performance in terms of relative textual *Improvement*. The *I*-measure is calculated by carrying out a 3-way alignment between the original, hypothesis and reference texts and classifying each token according to an extended version of the Writer-Annotator-System (WAS) evaluation scheme (Chodorow et al. 2012). This ultimately enables the calculation of accuracy, which Felice and Briscoe (2015) modify to weight TPs and FPs differently to more intuitively reward or punish a system. Having calculated a weighted accuracy score for a system, a baseline weighted accuracy score is computed in the same manner using a copy of the original text as the hypothesis. The difference between these scores is then normalised to fall between -1 and 1, where $I < 0$ indicates text degradation and $I > 0$ indicates text improvement.

GMEG. The GMEG metric (Napoles, Nădejde, and Tetreault 2019) is an ensemble metric that was designed to correlate with human judgements on three different datasets. It was motivated by the observation that different metrics correlate very differently with human judgements in different domains, and so a better metric would be more consistent. As an ensemble metric, GMEG depends on features (e.g. precision and recall) from several other metrics, including M^2, ERRANT, GLEU, and the I-measure (73 features in total). The authors then use these features to train a ridge regression model that was optimised to predict the human scores for different systems.

GoToScorer. The GoToScorer (Gotou et al. 2020) was motivated by the observation that some errors are more difficult to correct than others yet all metrics treat them equally. The GoToScorer hence models error difficulty by weighting edits according to how many different systems were able to correct them; e.g., edits that were successfully corrected by all systems would yield a smaller reward than those successfully corrected by fewer systems. Although this methodology confirmed the intuition that some errors types were easier to correct than others, e.g. spelling errors (easy) vs. synonym errors (hard), one disadvantage of this approach is that the difficulty weights depend entirely on the type and number of systems involved. Consequently, results do not generalise well and error difficulty (or gravity) remains an unsolved problem.

SERCL/SERRANT. SERCL (Choshen et al. 2020) is not a metric *per se*, but rather a method of automatically classifying grammatical errors by their syntactic properties using the Universal Dependencies formalism (Nivre et al. 2020). It is hence similar to ERRANT except it can more easily support other languages. The main disadvantage of SERCL is that it is not always meaningful to classify errors entirely based on their syntactic properties, e.g. spelling and orthography errors, and some error types are not very informative, e.g. "VERB→ADJ". SERRANT (Choshen et al. 2021) is hence a compromise that attempts to combine the advantages of both SERCL and ERRANT.

PT-M^2. The pretraining-based MaxMatch (PT-M^2) metric (Gong et al. 2022) is a hybrid metric that combines traditional edit-based metrics, such as M^2, with recent pretraining-based metrics, such as BERTScore (Zhang et al. 2020). The main advantage of pretraining-based metrics over edit-based metrics is that they are more capable of measuring the semantic similarity between pairs of sentences, rather than just comparing edits. Since Gong et al. (2022) found that off-the-shelf pretraining metrics correlated poorly with human judgements on GEC at the sentence level, they instead proposed measuring performance at the edit level. This approach ultimately produced the highest correlation with human judgements on the CoNLL-2014 test set to date, but should be considered with caution, as Hanna and Bojar (2021) also highlight some of the limitations of pretraining metrics and cite sources that claim correlation with human judgements may not be the best way to evaluate a metric (see Section 6.5).

6.4.2 Reference-less Metrics.

GBMs. The first work to explore the idea of a reference-less metric for GEC (Napoles, Sakaguchi, and Tetreault 2016) was inspired by similar work on *quality estimation* in machine translation (e.g. Specia et al. (2020)). Specifically, the authors proposed three Grammaticality-Based Metrics (GBMs) that either use a benchmark GEC system to count the errors in the output produced by other GEC systems or else predict a

grammaticality score using a pretrained ridge regression model (Heilman et al. 2014). The main limitation of these metrics is that they i) require an existing GEC system to evaluate other GEC systems and ii) are insensitive to changes in meaning. The authors thus proposed interpolating reference-less metrics with other reference-based metrics.

GFM. Asano, Mizumoto, and Inui (2017) extended the work on GBMs by introducing three reference-less metrics for Grammaticality, Fluency and Meaning preservation (GFM). Specifically, the Grammaticality metric combines Napoles, Sakaguchi, and Tetreault's 2016 GBMs into a single model, the Fluency metric computes a score using a language model, and the Meaning preservation metric computes a score using the METEOR metric from machine translation (Denkowski and Lavie 2014). A weighted linear sum of the three scores is then used as the final score. The main weaknesses of the GFM metric are that the Grammaticality and Fluency metrics suffer from the same limitations as GBMs, and the Meaning preservation metric only models shallow text similarity in terms of overlapping content words.

USim. The USim metric (Choshen and Abend 2018c) was motivated by the fact that no other metric takes deep semantic similarity into account and it is possible that a GEC system might change the intended meaning of the original text; e.g., by inserting/deleting 'not' or replacing a content word with an incorrect synonym. It is calculated by first automatically annotating the original and hypothesis texts as semantic graphs using the UCCA semantic scheme (Abend and Rappoport 2013) and then measuring the overlap between the graphs (in terms of matching edges) as an F-score. USim was thus designed to operate as a complementary metric to other metrics.

SOME. Sub-metrics Optimised for Manual Evaluation (SOME) (Yoshimura et al. 2020) is an extension of GFM that was designed to optimise each Grammaticality, Fluency and Meaning preservation metric to more closely correlate with human judgements. The authors achieved this by annotating the system output of five recent systems on a 5-point scale for each metric and then fine-tuning BERT (Devlin et al. 2019) to predict these human scores. This differs from GFM in that GFM was fine-tuned to predict the human ranking of different systems rather than explicit human scores. While the authors found SOME correlates more strongly with human judgements than GFM, both metrics nevertheless suffer from the same limitations.

Scribendi Score. The Scribendi Score (Islam and Magnani 2021) was designed to be simpler than other reference-less metrics in that it requires neither an existing GEC system nor fine-tuning. Instead, it calculates an absolute score (1=positive, -1=negative, 0=no change) from a combination of language model perplexity (GPT2: Radford et al. (2019)) and sorted token/Levenshtein distance ratios, which respectively ensure that i) the corrected sentence is more probable than the original and ii) both sentences are not significantly different from each other. While it is intuitive that these scores correlate with the grammaticality of a sentence, they are not, however, a robust way of evaluating a GEC system. For example, the sentence 'I saw *the* cat" is more probable than "I saw *a* cat" in GPT2 (160.8 vs 156.4), and both sentences are moreover very similar, yet we would not want to always reward this as a valid correction since both sentences are grammatical. We observe the same effect in "I ate the cake." (130.2) vs. "I ate the pie." (230.7) and so conclude that the Scribendi Score is highly likely to erroneously reward false positives.

IMPARA. The Impact-based Metric for GEC using Parallel data (IMPARA) (Maeda, Kaneko, and Okazaki 2022) is a hybrid reference-based/reference-less metric that requires parallel data to train an edit-based quality estimation and semantic similarity model, but can be used as a reference-less metric after training. It is sensitive to the corpus it is trained on (i.e., it does not generalise well to unseen domains) but shows comparable or better performance to SOME in terms of correlation with human judgements. Its main advantage is that it only requires parallel data for training (i.e., not system output or human judgements), but its main disadvantage is that IMPARA scores are not currently interpretable by humans.

6.5 Metric Reliability

Given the number of metrics that have been proposed, it is natural to wonder which metric is best. This is not straightforward to answer, however, as all metrics have different strengths and weaknesses. There has nevertheless been a great deal of work based on the assumption that the "best" metric is the one that correlates most closely with ground-truth human judgements.

With this in mind, the first work to compare metric performance with human judgements was by Napoles et al. (2015) and Grundkiewicz, Junczys-Dowmunt, and Gillian (2015), who independently collected human ratings for the 13 system outputs from the CoNLL-2014 shared task (including the unchanged original text) using the Appraise evaluation framework (Federmann 2010) commonly used in MT. This framework essentially asks humans to rank randomly chosen samples of 5 system outputs (ties are permitted) in order to build up a collection of pairwise judgements that can be used to extrapolate an overall system ranking. A metric can then be judged in terms of how well it correlates with this extrapolated ranking. The judgements collected by Grundkiewicz, Junczys-Dowmunt, and Gillian (2015) in particular proved especially influential (their dataset was much larger than Napoles et al. (2015)) and were variously used to justify GLEU as a better metric than M^2 (Napoles et al. 2015; Napoles, Sakaguchi, and Tetreault 2016; Sakaguchi et al. 2016) and motivate almost all reference-less metrics to date (except USIM).

Unfortunately however, this methodology was later found to be problematic and many of the conclusions drawn using these datasets were thrown into doubt. Notable observations included:

- The correlation coefficients reported by Napoles et al. (2015) and Grundkiewicz, Junczys-Dowmunt, and Gillian (2015) were very different even though they essentially carried out the same experiment (albeit on different samples) (Choshen and Abend 2018a).
- This method of human evaluation was abandoned in machine translation due to unreliability (Choshen and Abend 2018a; Graham, Baldwin, and Mathur 2015).
- Chollampatt and Ng (2018c) found no evidence of GLEU being a better metric than M^2 for ranking systems.

Choshen and Abend (2018a) surmise that one of the reasons these metric correlation experiments proved unreliable is that rating sentences for grammaticality is a highly subjective task which often shows very low inter-annotator agreement (IAA); e.g. it is

difficult to determine whether a sentence containing one major error should be considered "more grammatical" than a sentence containing two minor errors.

Napoles, Nădejde, and Tetreault (2019) nevertheless carried out a follow-up study which not only used a continuous scale to judge sentences (rather than rank them) (Sakaguchi and Van Durme 2018), but also collected judgements on all pairs of sentences to overcome sampling bias. They furthermore reported results on different datasets from different domains, rather than just CoNLL-2014, in an effort to determine the most generalisable metric. Their results, partially recreated in Table 7, hence found that dataset does indeed have an effect on metric performance, most likely because different error type distributions are judged inconsistently by humans. In fact, although Napoles, Nădejde, and Tetreault (2019) reported very high IAA at the corpus level (0.9-0.99 Pearson/Spearman), IAA at the sentence level was still low to average (0.3-0.6 Pearson/Spearman).

	FCE		Wiki		Yahoo	
Metric	r	ρ	r	ρ	r	ρ
ERRANT $F_{0.5}$	**0.919**	**0.887**	0.401	0.555	0.532	0.601
GLEU	0.838	0.813	0.426	0.538	0.740	0.775
I-measure	0.819	0.839	**0.854**	**0.875**	**0.915**	**0.900**
M^2 $F_{0.5}$	0.860	0.849	0.346	0.552	0.580	0.699

Table 7
Pearson r and Spearman ρ correlation coefficients for different metrics across three different datasets. This is a subset of the results reported in Napoles, Nădejde, and Tetreault (2019) Table 8.

Ultimately, although ground-truth human judgements may be an intuitive way to benchmark metric performance, they are also highly subjective and should be considered with caution. Nothing demonstrates this sentiment better than the conclusions drawn about the I-measure, which was initially found to have a weak negative correlation with human judgements (Napoles et al. 2015; Grundkiewicz, Junczys-Dowmunt, and Gillian 2015; Sakaguchi et al. 2016), subsequently found to have good correlation at the sentence level (Napoles, Sakaguchi, and Tetreault 2016) and finally considered the best singular metric across multiple domains (Napoles, Nădejde, and Tetreault 2019). Reliable methods of evaluating automatic metrics thus remain an active area of research.

6.6 Evaluation Best Practices

A common pitfall for new researchers in GEC concerns which metric to use with which dataset; for example the M^2 scorer with JFLEG, or the I-measure with BEA-2019. While there is no empirical reason to prefer one metric over another, in practice, the most popular GEC test sets are almost always evaluated with a single, specific metric:

- CoNLL-2014 is evaluated with the M^2 scorer
- JFLEG is evaluated with GLEU
- BEA-2019 is evaluated with ERRANT

This choice of experimental setup is largely motivated by historical reasons (e.g., GLEU and ERRANT did not exist during CoNLL-2014), but has nevertheless persisted

in order to ensure fair comparison with subsequent work. One particularly common mistake is to evaluate CoNLL-2014 with ERRANT or BEA-2019 with the M^2 scorer because both metrics return an F-score, yet M^2 $F_{0.5}$ is not equivalent to ERRANT $F_{0.5}$ (Bryant, Felice, and Briscoe 2017). It is thus imperative that a dataset be evaluated with its associated metric in order to facilitate a meaningful comparison.

6.6.1 Caveats.

Despite this convention, it is also important to highlight the limitations of this setup, as it is not always desirable to optimise different systems for different test sets using different metrics. Instead, we should remember that the ultimate goal of GEC is to build systems that generalise well, and so we should not place too much emphasis on specific experimental configurations. It is with this in mind that Mita et al. (2019) recommend evaluating on multiple corpora in order to reveal any systematic biases towards particular domains or user demographics, while Napoles, Nădejde, and Tetreault (2019) recommend evaluating using their trained metric that was designed to be less sensitive to dataset biases. These approaches thus add greater confidence that a model is versatile and does not overfit to a specific type of input.

6.6.2 Recommendations.

In light of the confusion surrounding different experimental setups, we make the following recommendations for ensuring a meaningful comparison in English GEC evaluation. This is not an exhaustive list, but we attempt to summarise the current standard experimental setups that facilitate the most informative comparison with previous work.

1. *Evaluate on the BEA-2019 test set using ERRANT.*
   The BEA-2019 test set is one of the most diverse test sets that contains texts from the full range of learner backgrounds and ability levels on a wide range of topics. This makes it a good benchmark for system robustness and generalisability. It is also the official test set of the most recent shared task.
2. *Evaluate on the CoNLL-2014 test set using the M^2 scorer.*
   The CoNLL-2014 test set is one of the most well-known test sets that has been widely used to benchmark progress in the field; it is thus an important indicator of system performance. It is also the official test set of the second most recent shared task.
3. *Evaluate on the GMEG and/or CWEB test sets using ERRANT.*
   One of the main limitations of the BEA-2019 and CoNLL-2014 test sets is that they mainly represent non-native language learners. It can therefore be beneficial to evaluate on native speaker errors in GMEG and CWEB to obtain a more complete picture of system generalisability.
4. *Evaluate on JFLEG using GLEU.*
   The main reason to evaluate on JFLEG is to test systems on more complex fluency edits rather than minimal edits. Not all edits in JFLEG are fluency edits however, and the test set is very small, so researchers have seldom reported GLEU on JFLEG in recent years (Gong et al. 2022).

Ultimately, robust evaluation is rarely as straightforward as directly comparing one number against another, and it is important to consider, for example, whether a model has been trained/fine-tuned on in-domain data, optimised for a specific metric, or only evaluated on a specific target test set. Each of these factors impacts how a score should be interpreted, especially in relation to previous work, and there is a real danger of rewarding a highly-optimised, specialised system, over a lower-scoring but more versatile system that may actually be more desirable.

7. System Comparison

In this section, we compare the most recent state-of-the-art systems from the past couple of years and comment on the innovations that led to them performing better than previous work. The full list of systems we compare is shown in Table 8. For a comparison of systems between 2014-2020, we refer the reader to Wang et al. (2021, Table 7).

7.1 System Description

We first note that many of the systems in Table 8 are extensions of 3 other systems: Omelianchuk et al. (2020), Sun et al. (2021), and Kiyono et al. (2019). Specifically, Omelianchuk et al. (2020) built a sequence tagging model (Section 3.4) using a pre-trained language model (e.g. BERT) and 9 million synthetic sentence pairs, Sun et al. (2021) used a rule-based approach to generate 300 million synthetic sentence pairs (Section 5.1.1) to train a modified BART model which contains 12 encoders and 2 decoders, and Kiyono et al. (2019) used 70 million synthetic sentence pairs generated through back-translation (Section 5.1.2) to train a Transformer-big model.

Many of these system specifically build on top of Omelianchuk et al. (2020), including systems from Sorokin (2022); Lai et al. (2022); Parnow, Li, and Zhao (2021); Yasunaga, Leskovec, and Liang (2021). Specifically, Sorokin (2022) and Tarnavskyi, Chernodub, and Omelianchuk (2022) upgraded the pre-trained language model from base to large (e.g., RoBERTa-base vs. RoBERTa-large) and employed an additional mechanism to select the final edits by means of edit-scoring or majority voting (VT) respectively. Parnow, Li, and Zhao (2021) and Lai et al. (2022) address the problem of edit interdependence, i.e. when the correction of one error depends on another, by means of GANs and multi-turn training respectively. Yasunaga, Leskovec, and Liang (2021) applied the break-it-fix-it (BIFI) framework (Yasunaga and Liang 2021) to Omelianchuk et al. (2020) (Section 3.5) to gradually train a system that iteratively generates and learns from more realistic synthetic data. In contrast, Sun and Wang (2022) add a single hyperparameter to Sun et al. (2021) to control the trade-off between precision and recall (PRT), Kaneko et al. (2020) incorporate BERT into Kiyono et al. (2019) (Section 3.3.3), and Mita et al. (2020) applied a self-refinement data augmentation strategy to Kiyono et al. (2019) (Section 5.2).

System	Synthetic Sents	Corpora	Pre-trained Model	Architecture	Techniques	CoNLL14 M2	BEA19 ERRANT
Qorib, Na, and Ng (2022)	-	W (dev)	Various[1]	T5-large, RoBERTa-base, XLNet-base, Transformer-big	SC	69.5	79.9
Lai et al. (2022)	9m	N+F+L+W	RoBERTa, XLNet	RoBERTa-base, XLNet-base	ENS+PRT+MTD	67.0	77.9
Sorokin (2022)	9m	cL+N+F+W	RoBERTa	RoBERTa-large	RE+MTD	64.0	77.1
Tarnavskyi, Chernodub, and Omelianchuk (2022)	-	N+F+L+W	RoBERTa, XLNet, DeBERTa	RoBERTa-large, XLNet-large, DeBERTa-large	VT+PRT+MTD	65.3	76.1
Rothe et al. (2021)	-	cL	T5-xxl	T5-xxl	-	68.9	75.9
Sun and Wang (2022)	300m	N+F+L+W	BART	BART (12+2)	PRT	-	75.0
Stahlberg and Kumar (2021)	546m	F+L+W	-	Transformer-big	ENS	68.3	74.9
Omelianchuk et al. (2020)	9m	N+F+L+W	BERT, RoBERTa, XLNet	BERT-base, RoBERTa-base, XLNet-base	ENS+PRT+MTD	66.5	73.7
Lichtarge, Alberti, and Kumar (2020)	340m	F+L+W	-	Transformer-big	ENS	66.8	73.0
Zhang et al. (2022c)	-	cL+N+F+W	BART	BART-large	-	67.6	72.9
Sun et al. (2021)	300m	N+F+L+W	BART	BART (12+2)	-	66.4	72.9
Yasunaga, Leskovec, and Liang (2021)	9m	N+F+L+W	XLNet	XLNet-base	PRT+MTD	65.8	72.9
Parnow, Li, and Zhao (2021)	9m	N+F+L+W	XLNet	XLNet-base	PRT+MTD	65.7	72.8
Yuan et al. (2021)	-	N+F+L+W +CLC	ELECTRA	Multi-encoder, Transformer-base	RE	63.5	70.6
Stahlberg and Kumar (2020)	346m	F+L+W	-	Seq2Edits (modified Transformer-big)	ENS+RE	62.7	70.5
Kaneko et al. (2020)	70m	N+F+L+W	-	Transformer-big	ENS+RE	65.2	69.8
Mita et al. (2020)	70m	N+F+L+W	-	Transformer-big	ENS+RE	63.1	67.8
Chen et al. (2020a)	260m	N+F+L+W	RoBERTa	Transformer-big	-	61.0	66.9
Katsumata and Komachi (2020)	-	N+F+L+W	BART	BART-large	ENS	63.0	66.1

[1] Combines Rothe et al. (2021); Omelianchuk et al. (2020); Kiyono et al. (2019); Grundkiewicz, Junczys-Dowmunt, and Heafield (2019); Choe et al. (2019).

Table 8
Top-performing systems since 2020. The symbols in the Corpora column are N: NUCLE, F: FCE, L: Lang-8, W: W&I, cL: cLang-8, CLC: Cambridge Learner Corpus, and C: CWEB. The symbols in the Techniques column are ENS: ensemble, MTD: multi-turn decoding, PRT: precision-recall trade-off, RE: re-ranking, SC: system combination, and VT: voting combination.

Other systems include Katsumata and Komachi (2020) and Rothe et al. (2021), who respectively explored the effectiveness of using pre-trained BART (Lewis et al. 2020) and T5 (Raffel et al. 2020) as the base model for GEC; Zhang et al. (2022c) subsequently extended Katsumata and Komachi (2020) by adding syntactic information (Section 3.3.3). Chen et al. (2020a) and Yuan et al. (2021) meanwhile both combined error detection with error correction by respectively constraining the output of a GEC system based on a separate GED system and jointly training GED as an auxiliary task (Section 4.3). Stahlberg and Kumar (2020) proposed a seq2edit approach that explicitly predicts a sequence of tuple edit operations to apply to an input sentence (Section 3.4), while Stahlberg and Kumar (2021) developed a method to generate a specific type of error in a sentence (given a clean sentence and an error tag), which could be used to generate synthetic datasets that more closely match the error distribution in a real corpus (Section 5.1.2). Finally, Lichtarge, Alberti, and Kumar (2020) used delta-log-perplexity to weight the contribution of each sentence in the training set towards overall model performance, downweighting those that added the most noise (Section 5.2), and Qorib, Na, and Ng (2022) used a binary classifier based on logistic regression to combine multiple GEC systems using only the output from each individual component system.

7.2 Analysis

Despite all these enhancements, we first observe that it is very difficult to draw conclusions about the efficacy of different techniques in Table 8, because different systems were trained using different amounts/types of data (both real and artificial) and developed using different pre-trained models and performance-boosting techniques. Consequently, the systems are rarely directly comparable and we can only infer the relative advantages of different approaches from the wider context. With this in mind, the general trend in the past couple of years has been to scale models up using i) more artificial data, ii) multiple pre-trained models/architectures, and iii) multiple performance-boosting techniques.

In terms of artificial data, the trend is somewhat mixed, as on the one hand, Stahlberg and Kumar (2021) introduced a system trained on more than half a billion synthetic sentences, but on the other hand, they were still outperformed by systems that used orders of magnitude less data (Lai et al. 2022; Tarnavskyi, Chernodub, and Omelianchuk 2022). This pattern has been consistent for several years now and reveals a delicate trade-off between artificial data quantity and quality. There is ultimately no clear relationship between data quantity and performance, and some systems still achieve competitive performance without artificial data (Rothe et al. 2021; Yuan et al. 2021; Katsumata and Komachi 2020).

The use of several pre-trained model architectures, however, tells a different story and it is generally the case that using multiple architectures improves performance: the top 3 latest state-of-the-art systems all use at least 2 different pre-trained models (Qorib, Na, and Ng 2022; Lai et al. 2022; Tarnavskyi, Chernodub, and Omelianchuk 2022). This suggests that different pre-training tasks capture different aspects of natural language that complement each other in different ways in GEC. In contrast, approaches that rely on a single pre-trained model typically perform slightly worse than those that combine architectures, although it is worth keeping in mind that there is also a trade-off between model complexity and run-time which is seldom reported (Omelianchuk et al. 2020; Sun et al. 2021).

Finally, adding more performance-boosting techniques also tends to result in better performance, and the systems that incorporate the most techniques typically score

highest. Among these techniques, the use of model ensembling or system combination (Section 4.2) mitigates the instability of neural models and allows a final system to make use of the strengths of several other systems. However, this comes at a cost to model complexity and run-time.

8. Future Challenges

While much progress has been made in the past decade, several important challenges remain (Qorib and Ng 2022). This section highlights some of them and offers suggestions for future work.

Domain Generalisation. Robustness is an important attribute of any NLP system. In the case of GEC, we not only want our systems to work well for language learners, but also native speakers in different domains such as business emails, literature and instruction manuals. Some efforts have been made in this direction, such as the native web texts in CWEB (Flachs et al. 2020), scientific articles in AESW (Daudaravicius et al. 2016) and conversational dialog in ErAConD (Yuan et al. 2022), but more effort is needed to create new corpora that represent a wider variety of domains. This is important because previous research has shown that systems that perform well in one domain do not necessarily perform well in other domains (Napoles, Nădejde, and Tetreault 2019).

Personalised Systems. Related to domain generalisation is the fact that system performance is also tied to the profiles of the users in the training data. For example, a system trained on L2 English data produced by advanced L1 Spanish learners is unlikely to perform as well on L2 English data produced by beginner L1 Japanese learners because of the mismatch in ability level and first language. It is thus important to develop corpora and tools that can adapt to different users, since different ability levels and L1s can significantly affect the distribution of errors that authors are likely to make (Nadejde and Tetreault 2019).

Feedback Comment Generation. GEC systems are currently trained to correct errors without explaining why a correction was needed. This is insufficient in an educational context however, where it is desirable for a system to explain the cause of an error such that a user may learn from it and not make the same mistake again. Resources have begun to emerge to support this endeavour, but much more work is needed to generate robust feedback comments to support explainable GEC (Nagata 2019; Nagata, Inui, and Ishikawa 2020; Hanawa, Nagata, and Inui 2021; Nagata et al. 2021).

Model Interpretability. Related to feedback generation, it is also important that model output is interpretable by humans. For example, although a system may make a prediction with high confidence, there is no guarantee that the prediction will be consistent with human intuition. Researchers have thus begun to build systems that estimate the quality of model output in an effort to provide more confidence that a given prediction is correct (Chollampatt and Ng 2018b; Liu et al. 2021). Similarly, Kaneko et al. (2022) propose an example-based approach, where a model additionally outputs similar corrections in different contexts in order to add credibility to the notion that the model truly understood the error.

Semantic Errors. One of the areas where state-of-the-art systems still underperform is semantic errors, which include complex phenomenon such as collocations, idioms,

multi-word expressions and fluency edits. A lot of work in GEC has focused on correcting function word errors, which typically have small confusion sets and comprise a majority of error types, but this does not mean we can neglect the correction of content word errors. Although there has been some work on correcting collocations (Kochmar and Briscoe 2014; Herbelot and Kochmar 2016) and multi-word expressions (Mizumoto, Mita, and Matsumoto 2015; Taslimipoor, Bryant, and Yuan 2022), semantic errors remain a notable area in which GEC systems could improve.

Contextual GEC. To date, most GEC systems operate at the sentence level, and so do not perform well on errors that require cross-sentence context or document-level understanding. Although work has already been done to incorporate multi-sentence context into GEC systems (Chollampatt, Wang, and Ng 2019; Yuan and Bryant 2021; Mita et al. 2022), almost all current datasets expect sentence tokenised input and so do not facilitate multi-sentence evaluation. Paragraph or document-level datasets, like in the Arabic QALB shared tasks (Mohit et al. 2014; Rozovskaya et al. 2015), should thus be developed to encourage contextual GEC in the future.

System Combination. Although much recent work focuses on NMT for GEC, this does not mean that other approaches have nothing to offer. Work on system combination has shown that systems built with different approaches have complementary strengths and weaknesses such that a combined system can achieve significantly improved performance (Susanto, Phandi, and Ng 2014; Han and Ng 2021; Lin and Ng 2021; Qorib, Na, and Ng 2022). Better understanding of these strengths and weaknesses, and when and how to combine approaches are promising areas of research. One tool is ALLECS (Qorib, Moon, and Ng 2023), which is a web-interface tool to produce text corrections using GEC system combination methods.

Training Data Selection. Current state-of-the-art systems rely on pre-training on a massive amount of synthetic parallel corpora, however this is both computationally-expensive and not environmentally friendly. It is also questionable whether so much training data is really necessary, as humans are not exposed to training data on such a massive scale, yet can still correct errors without issue. A more economical approach to effective training data selection is thus an important research question that will go a long way towards reducing training time and developing more efficient GEC systems (Lichtarge, Alberti, and Kumar 2020; Takahashi, Katsumata, and Komachi 2020; Mita and Yanaka 2021; Rothe et al. 2021).

Unsupervised Approaches. The dependency on parallel corpora (both real and synthetic) is a major limiting factor in GEC system development, in that it is both laborious and time-consuming to train human annotators to manually correct errors, and also surprisingly difficult to generate high-quality synthetic errors that reliably imitate human error patterns. It is furthermore noteworthy that humans can correct errors without access to a large corpus of erroneous examples and instead rely on their knowledge of grammatical language in order to detect and correct mistakes. It should thus be intuitive that a GEC system might be able to do the same by interpreting deviations from grammatical data as anomalies that need to be corrected. The success of such an unsupervised approach would significantly hasten the development of multilingual GEC systems and also eliminate the need to compile parallel corpora.

Multilingual GEC. Although most work on GEC has focused on English, work on other languages is also beginning to take off as new resources become available; e.g. in German (Boyd 2018), Russian (Rozovskaya and Roth 2019) and Czech (Náplava et al. 2022). While it is important to encourage research into GEC systems for specific languages, it is also important to remember that it is ultimately not scalable to build a separate system for every language. It is desirable to work towards a single multilingual system that can correct all languages simultaneously like in machine translation (Katsumata and Komachi 2020; Rothe et al. 2021).

Spoken GEC. Another aspect of GEC that has seldom been explored in the literature is that of spoken GEC. While progress has largely been hindered by a lack of available data, researchers have recently begun to build systems capable of detecting and correcting errors in learner speech (Knill et al. 2019; Caines et al. 2020; Kyriakopoulos, Knill, and Gales 2020; Lu, Gales, and Wang 2020; Lu, Bannò, and Gales 2022). Compared to text-based GEC, additional challenges include recognising non-native accented speech (possibly including non-standard pronunciation), disfluency detection, and utterance segmentation.

Improved Evaluation. Finally, robust evaluation of GEC system output is still an unsolved problem and current evaluation practices may actually hinder progress (Rozovskaya and Roth 2021). For example, almost all metrics to date require tokenised text, yet end-users require untokenised text, and so there is a disconnect between system capability and user expectation. Similarly, GEC systems are typically trained to output a single best correction for a sentence, yet end-users may prefer a short n-best list of possible corrections for each edit, like in most spellcheckers. Ultimately, alternative answers and untokenised text are not yet properly accounted for in GEC system evaluation, leaving room for new metrics to drive the field towards better practices.

9. Conclusion

In this survey paper, we set out to provide a comprehensive overview of the state of the art in the field of Grammatical Error Correction. Our main goal was to summarise the progress that has been made since Leacock et al. (2014) but also complement the work of Wang et al. (2021) with more in-depth coverage on various topics.

With this in mind, we first explored the nature of the task and illustrated the inherent difficulties in defining an error according to the perceived communicative intent of the author. We next alluded to how these difficulties can manifest in human-annotated corpora, before introducing the most commonly used benchmark corpora for English, several less commonly used corpora for English, and new corpora for GEC systems in other languages, including Arabic, Chinese, Czech, German and Russian. Research into GEC for non-English languages has begun to take off in the last couple of years and will no doubt continue to grow in the future.

We next characterised the evolution of approaches to GEC, from error-type specific classifiers to state-of-the-art NMT and edit-based sequence-labelling, and summarised some of the additional supplementary techniques that are commonly used to boost performance, such as re-ranking, multi-task learning and iterative decoding. We also described different methods of artificial data generation and augmentation, which have become core components of recent GEC systems, but also drew attention to the benefits of low-resource GEC systems that may be less resource intensive and more easily extended to other languages.

Robust evaluation is still an unsolved problem in GEC, but we introduced the most commonly used metrics in the field, along with their strengths and weaknesses, and listed previous attempts at both reference-based and reference-less metrics that were designed to overcome various shortcomings. We furthermore highlighted the difficulty in correlating human judgements with metric performance in light of the highly subjective nature of the task.

Finally, we provided an analysis of very recent progress in the field, including making observations about which techniques/resources seemed to perform best (particularly in the context of model efficiency), before concluding with several possibilities for future work. We hope that this survey will serve as comprehensive resource for researchers who are new to the field or who want to be kept apprised of recent developments.

References

Abend, Omri and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.

Alikaniotis, Dimitris and Vipul Raheja. 2019. The unreasonable effectiveness of transformer language models in grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–133.

Alsufieva, Anna A., Olesya V. Kisselev, and Sandra G. Freels. 2012. Results 2012: Using flagship data to develop a russian learner corpus of academic writing. *Russian Language Journal*, 62:79–105.

Asano, Hiroki, Masato Mita, Tomoya Mizumoto, and Jun Suzuki. 2019. The AIP-tohoku system at the BEA-2019 shared task. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 176–182.

Asano, Hiroki, 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.

Awasthi, Abhijeet, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Belkebir, Riadh and Nizar Habash. 2021. Automatic error type annotation for Arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606.

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

Berend, Gábor, Veronika Vincze, Sina Zarrieß, and Richárd Farkas. 2013. LFG-based features for noun number and article grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 62–67.

van den Bosch, Antal and Peter Berck. 2013. Memory-based grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 102–108.

Boyd, Adriane. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84.

Boyd, Adriane, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*,

pages 1281–1288.
Brockett, Chris, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256.
Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
Bryant, Christopher. 2019. *Automatic annotation of error types for grammatical error correction*. Ph.D. thesis, University of Cambridge.
Bryant, Christopher and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253.
Bryant, Christopher and Mariano Felice. 2016. Issues in preprocessing current datasets for grammatical error correction. Technical Report UCAM-CL-TR-894, University of Cambridge, Computer Laboratory.
Bryant, Christopher, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.
Bryant, Christopher, 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.
Bryant, Christopher 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.
Burstein, Jill, Martin Chodorow, and Claudia Leacock. 2003. Criterion^{sm} online essay evaluation: An application for automated evaluation of student essays. In *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence*, pages 3–10.
Caines, Andrew, Christian Bentz, Kate Knill, Marek Rei, and Paula Buttery. 2020. Grammatical error detection in transcriptions of spoken English. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2144–2162.
Cao, Hannan, Wenmian Yang, and Hwee Tou Ng. 2021. Grammatical error correction with contrastive learning in low error density domains. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4867–4874.
Chen, Mengyun, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020a. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169.
Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607.
Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.
Chodorow, Martin, Markus Dickinson, Ross Israel, and Joel Tetreault. 2012. Problems in evaluating grammatical error detection systems. In *Proceedings of COLING 2012*, pages 611–628.

Chodorow, Martin, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30.

Choe, Yo Joong, 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 Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227.

Chollampatt, Shamil and Hwee Tou Ng. 2017. Connecting the dots: Towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 327–333.

Chollampatt, Shamil 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 (AAAI-18)*, pages 5755–5762.

Chollampatt, Shamil and Hwee Tou Ng. 2018b. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2528–2539.

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

Chollampatt, Shamil, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2768–2774.

Chollampatt, Shamil, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445.

Choshen, Leshem and Omri Abend. 2018a. Automatic metric validation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1372–1382.

Choshen, Leshem and Omri Abend. 2018b. 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.

Choshen, Leshem and Omri Abend. 2018c. Reference-less measure of faithfulness for grammatical error correction. 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 124–129.

Choshen, Leshem, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. Classifying syntactic errors in learner language. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107.

Choshen, Leshem, Matanel Oren, Dmitry Nikolaev, and Omri Abend. 2021. SERRANT: a syntactic classifier for english grammatical error types. *arXiv*, 2104.02310.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv*, 2204.02311.

Council of Europe. 2001. *Common European Framework of Reference for Languages:*

Learning, Teaching, Assessment.
Crowston, Kevin. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221.
Dahlmeier, Daniel and Hwee Tou Ng. 2011a. Correcting semantic collocation errors with L1-induced paraphrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.
Dahlmeier, Daniel and Hwee Tou Ng. 2011b. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 915–923.
Dahlmeier, Daniel and Hwee Tou Ng. 2012a. A beam-search decoder for grammatical error correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578.
Dahlmeier, Daniel and Hwee Tou Ng. 2012b. 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.
Dahlmeier, Daniel, 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.
Dale, Robert, 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.
Daudaravicius, Vidas, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62.
De Felice, Rachele. 2008. *Automatic error detection in non-native English*. Ph.D. thesis, Oxford University.
De Felice, Rachele and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176.
Denkowski, Michael and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380.
Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
Edunov, Sergey, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.
Ehsan, Nava and Heshaam Faili. 2013. Grammatical and context-sensitive error correction using a statistical machine translation framework. *Software: Practice and Experience*, 43(2):187–206.
Federmann, Christian. 2010. Appraise: An open-source toolkit for manual phrase-based evaluation of translations. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.
Felice, Mariano. 2016. *Artificial error generation for translation-based grammatical error correction*. Ph.D. thesis, University of Cambridge.
Felice, Mariano 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.
Felice, Mariano, 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.

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

Felice, Mariano and Zheng Yuan. 2014b. To err is human, to correct is divine. *XRDS*, 21(1):22–27.

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

Flachs, Simon, Ophélie Lacroix, and Anders Søgaard. 2019. Noisy channel for low resource grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 191–196.

Flachs, Simon, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478.

Flachs, Simon, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122.

Gamon, Michael. 2010. Using mostly native data to correct errors in learners' writing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 163–171.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Ge, Tao, 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.

Geertzen, Jeroen, Theodora Alexopoulou, and Anna Korhonen. 2013. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In *Selected Proceedings of the 31st Second Language Research Forum (SLRF)s*.

Gehring, Jonas, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135.

Gong, Peiyuan, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. `http://www.deeplearningbook.org`.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27.

Gotou, Takumi, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095.

Graham, Yvette, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1183–1191.

Granger, Sylviane. 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.

Grundkiewicz, Roman 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 – Lecture Notes in Computer Science*, volume 8686, pages 478–490.

Grundkiewicz, Roman 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.

Grundkiewicz, Roman and Marcin Junczys-Dowmunt. 2019. Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363.

Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470.

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

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

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in english article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Han, Wenjuan and Hwee Tou Ng. 2021. Diversity-driven combination for grammatical error correction. In *Proceedings of ICTAI*.

Hanawa, Kazuaki, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730.

Hanna, Michael and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

Heilman, Michael, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180.

Herbelot, Aurélie and Ekaterina Kochmar. 2016. 'calling on the classical phone': a distributional model of adjective-noun errors in learners' English. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 976–986.

Hoang, Duc Tam, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an smt approach to grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoskere, Jayakumar. 2019. Everyday AI: beyond spell check, how Google Docs is smart enough to correct grammar. *Google Blogs*.

Hotate, Kengo, Masahiro Kaneko, and Mamoru Komachi. 2020. Generating diverse corrections with local beam search for grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2132–2137.

Htut, Phu Mon and Joel Tetreault. 2019. The unbearable weight of generating artificial errors for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 478–483.

Huang, Yan, Jeroen Geertzen, Rachel Baker, Anna Korhonen, and Theodora Alexopoulou. 2017. The EF Cambridge Open Language Database (EFCAMDAT): Information for Users. Technical report, Department of Theoretical and Applied Linguistics, University of Cambridge and EF Education First.

Islam, Md Asadul and Enrico Magnani. 2021. Is this the end of the gold standard? a straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing, pages 3009–3015.
Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.
Jensen, K., G. E. Heidorn, L. A. Miller, and Y. Ravin. 1983. Parse fitting and prose fixing: Getting a hold on ill-formedness. *American Journal of Computational Linguistics*, 9(3-4):147–160.
Ji, Jianshu, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762.
Jia, Zhongye, Peilu Wang, and Hai Zhao. 2013. Grammatical error correction as multiclass classification with single model. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 74–81.
Junczys-Dowmunt, Marcin and Roman Grundkiewicz. 2014. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33.
Junczys-Dowmunt, Marcin 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.
Junczys-Dowmunt, Marcin, 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.
Kalchbrenner, Nal and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.
Kaneko, Masahiro, Kengo Hotate, Satoru Katsumata, and Mamoru Komachi. 2019. TMU transformer system using BERT for re-ranking at BEA 2019 grammatical error correction on restricted track. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–212.
Kaneko, Masahiro, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254.
Kaneko, Masahiro, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187.
Kantor, Yoav, 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 BEA*, pages 139–148.
Katsumata, Satoru and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832.
Ke, Zixuan and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308.
Kiyono, Shun, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242.
Knill, K.M., M.J.F. Gales, P.P. Manakul, and A.P. Caines. 2019. Automatic grammatical error detection of non-native

spoken learner english. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8127–8131.

Kochmar, Ekaterina, Øistein Andersen, and Ted Briscoe. 2012. HOO 2012 error recognition and correction shared task: Cambridge University submission report. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 242–250.

Kochmar, Ekaterina and Ted Briscoe. 2014. Detecting learner errors in the choice of content words using compositional distributional semantics. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1740–1751.

Korre, Katerina, Marita Chatzipanagiotou, and John Pavlopoulos. 2021. ELERRANT: Automatic grammatical error type classification for Greek. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 708–717.

Koyama, Aomi, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. Comparison of grammatical error correction using back-translation models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135.

Koyama, Aomi, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 204–211.

Kunchukuttan, Anoop, Sriram Chaudhury, and Pushpak Bhattacharyya. 2014. Tuning a grammar correction system for increased precision. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 60–64.

Kwasny, Stan C. and Norman K. Sondheimer. 1981. Relaxation techniques for parsing grammatically ill-formed input in natural language understanding systems. *American Journal of Computational Linguistics*, 7(2):99–108.

Kyriakopoulos, Konstantinos, Kate M. Knill, and Mark J. F. Gales. 2020. Automatic detection of accent and lexical pronunciation errors in spontaneous non-native english speech. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3052–3056.

Lai, Shaopeng, Qingyu Zhou, Jiali Zeng, Zhongli Li, Chao Li, Yunbo Cao, and Jinsong Su. 2022. Type-driven multi-turn corrections for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3225–3236.

Leacock, Claudia, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners*.

Leacock, Claudia, Michael Gamon, and Chris Brockett. 2009. User input and interactions on Microsoft Research ESL assistant. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 73–81.

Lee, John. 2004. Automatic article restoration. In *Proceedings of the Student Research Workshop at HLT-NAACL 2004*, pages 31–36.

Lee, John and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182.

Lee, Kyusong and Gary Geunbae Lee. 2014. POSTECH grammatical error correction system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 65–73.

Lee, Lung-Hao, Yuen-Hsien Tseng, and Li-Ping Chang. 2018. Building a TOCFL learner corpus for Chinese grammatical error diagnosis. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Levenshtein, Vladimir I. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707–710.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

Li, Ruobing, 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 Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–167.

Lichtarge, Jared, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

Lichtarge, Jared, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301.

Lin, Ruixi and Hwee Tou Ng. 2021. System combination for grammatical error correction based on integer programming. In *Proceedings of RANLP*, pages 829–834.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Liu, Xiaodong, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very deep transformers for neural machine translation. *arXiv*, 2008.07772.

Liu, Zhenghao, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural quality estimation with multiple hypotheses for grammatical error correction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5441–5452.

Lu, Yiting, Stefano Bannò, and Mark Gales. 2022. On assessing and developing spoken 'grammatical error correction' systems. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 51–60.

Lu, Yiting, Mark J. F. Gales, and Yu Wang. 2020. Spoken language 'grammatical error correction'. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3840–3844.

Madnani, Nitin, Joel Tetreault, and Martin Chodorow. 2012. Exploring grammatical error correction with not-so-crummy machine translation. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53.

Maeda, Koki, Masahiro Kaneko, and Naoaki Okazaki. 2022. IMPARA: Impact-based metric for GEC using parallel data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588.

Malmi, Eric, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26.

Mita, Masato, Shun Kiyono, Masahiro Kaneko, Jun Suzuki, and Kentaro Inui. 2020. A self-refinement strategy for noise reduction in grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 267–280.

Mita, Masato, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models — is single-corpus evaluation enough? In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1309–1314.

Mita, Masato, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2022. Towards automated document revision: Grammatical error correction, fluency edits, and beyond. *arXiv*, 2205.11484.

Mita, Masato and Hitomi Yanaka. 2021. Do grammatical error correction models realize grammatical generalization? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4554–4561.

Mizumoto, Tomoya, 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.

Mizumoto, Tomoya, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155.

Mizumoto, Tomoya and Yuji Matsumoto. 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1133–1138.

Mizumoto, Tomoya, Masato Mita, and Yuji Matsumoto. 2015. Grammatical error correction considering multi-word expressions. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 82–86.

Mohit, Behrang, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47.

Muennighoff, Niklas, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning. *arXiv*, 2211.01786.

Nadejde, Maria and Joel Tetreault. 2019. Personalizing grammatical error correction: Adaptation to proficiency level and L1. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 27–33.

Nagata, Ryo. 2019. Toward a task of feedback comment generation for writing learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3206–3215.

Nagata, Ryo, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. Shared task on feedback comment generation for language learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324.

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

Náplava, Jakub and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356.

Náplava, Jakub, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. Czech Grammar Error Correction with a Large and Diverse Corpus. *Transactions of the Association for Computational Linguistics*, 10:452–467.

Napoles, Courtney, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Napoles, Courtney, 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.

Napoles, Courtney, Keisuke Sakaguchi, Matt Post, and Joel R. Tetreault. 2016. GLEU without tuning. *arXiv*, 1605.02592.

Napoles, Courtney, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Reference-less evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on*

Empirical Methods in Natural Language Processing, pages 2109–2115.

Napoles, Courtney, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: a fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234.

Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014a. 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.

Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014b. 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.

Ng, Hwee Tou, 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.

Nicholls, Diane. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 572–581.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

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

Opitz, David and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *J. Artif. Int. Res.*, 11(1):169–198.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Parnow, Kevin, Zuchao Li, and Hai Zhao. 2021. Grammatical error correction as GAN-like sequence labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Putra, Desmond Darma and Lili Szabó. 2013. UdS at CoNLL 2013 shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 88–95.

Qorib, Muhammad Reza, Geonsik Moon, and Hwee Tou Ng. 2023. ALLECS: A Lightweight Language Error Correction System. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

Qorib, Muhammad Reza, Seung-Hoon Na, and Hwee Tou Ng. 2022. Frustratingly easy system combination for grammatical error correction. In *Proceedings of the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Qorib, Muhammad Reza and Hwee Tou Ng. 2022. Grammatical error correction: Are we there yet? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2794–2800.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Raheja, Vipul and Dimitris Alikaniotis. 2020. Adversarial Grammatical Error Correction. In *Findings of the Association for Computational Linguistics: EMNLP*

2020, pages 3075–3087.

Rao, Gaoqi, Erhong Yang, and Baolin Zhang. 2020. Overview of nlptea-2020 shared task for chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35.

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

Rei, Marek and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191.

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

Reznicek, Marc, Anke Ludeling, Cedric Krummes, and Franziska Schwantuschke. 2012. Das FalkoHandbuch. Korpusaufbau und Annotationen Version 2.0.

van Rijsbergen, Cornelis J. 1979. *Information Retrieval*, 2nd edition.

Rosen, Alexandr. 2016. Building and using corpora of non-native Czech. In *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 80–87.

Rothe, Sascha, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707.

Rozovskaya, Alla, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015. The second QALB shared task on automatic text correction for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35.

Rozovskaya, Alla and Dan Roth. 2010a. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 961–970.

Rozovskaya, Alla and Dan Roth. 2010b. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 154–162.

Rozovskaya, Alla and Dan Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 924–933.

Rozovskaya, Alla and Dan Roth. 2013. Joint learning and inference for grammatical error correction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 791–802.

Rozovskaya, Alla and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

Rozovskaya, Alla and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Rozovskaya, Alla and Dan Roth. 2021. How good (really) are grammatical error correction systems? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2686–2698.

Rozovskaya, Alla, Dan Roth, and Vivek Srikumar. 2014. Correcting grammatical verb errors. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–367.

Sakaguchi, Keisuke, 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.

Sakaguchi, Keisuke and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218.

Sanh, Victor, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Šebesta, Karel. 2010. Korpusy češtiny a osvojování jazyka [Corpora of Czech and language acquistion]. *Studie z aplikované lingvistiky/Studies in Applied Linguistics*, 1:11–34.

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

Shannon, Claude E. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423.

Sorokin, Alexey. 2022. Improved grammatical error correction by ranking elementary edits. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429.

Specia, Lucia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764.

Stahlberg, Felix, Christopher Bryant, and Bill Byrne. 2019. Neural grammatical error correction with finite state transducers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4033–4039.

Stahlberg, Felix and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159.

Stahlberg, Felix and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47.

Sun, Chengjie, Xiaoqiang Jin, Lei Lin, Yuming Zhao, and Xiaolong Wang. 2015. Convolutional neural networks for correcting english article errors. In *Proceedings of the 4th National CCF Conference on Natural Language Processing and Chinese Computing*, pages 102–110.

Sun, Xin, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5937–5947.

Sun, Xin and Houfeng Wang. 2022. Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693.

Susanto, Raymond Hendy, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 951–962.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger,

editors, *Advances in Neural Information Processing Systems 27*. pages 3104–3112.
Syvokon, Oleksiy and Olena Nahorna. 2021. UA-GEC: grammatical error correction and fluency corpus for the ukrainian language. *arXiv*, 2103.16997.
Tajiri, Toshikazu, 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.
Takahashi, Yujin, Satoru Katsumata, and Mamoru Komachi. 2020. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32.
Tarnavskyi, Maksym, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852.
Taslimipoor, Shiva, Christopher Bryant, and Zheng Yuan. 2022. Improving grammatical error correction for multiword expressions. In *Proceedings of the 18th Workshop on Multiword Expressions (MWE 2022)*.
Tetreault, Joel, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358.
Tetreault, Joel R. and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872.
Trinh, Viet Anh and Alla Rozovskaya. 2021. New dataset and strong baselines for the grammatical error correction of Russian. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4103–4111.
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.
Wan, Zhaohong, Xiaojun Wan, and Wenguang Wang. 2020. Improving grammatical error correction with data augmentation by editing latent representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212.
Wang, Chuan, Ruobing Li, and Hui Lin. 2017. Deep Context Model for Grammatical Error Correction. In *Proceedings of the 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE 2017)*, pages 167–171.
Wang, Yiren, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *Proceedings of International Conference on Learning Representations*.
Wang, Yu, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Trans. Intell. Syst. Technol.*, 12(5).
Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.
Wei, Jason, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948.
White, Max and Alla Rozovskaya. 2020. A comparative study of synthetic data generation methods for grammatical error correction. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–208.
Xiang, Yang, Bo Yuan, Yaoyun Zhang, Xiaolong Wang, Wen Zheng, and Chongqiang Wei. 2013. A hybrid model for grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 115–122.
Xie, Ziang, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural Language Correction with Character-Based Attention. *arXiv*, 1603.09727.
Xie, Ziang, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language:

Diverse backtranslation for grammar correction. 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 619–628.

Xu, Shuyao, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158.

Yannakoudakis, Helen, 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.

Yannakoudakis, Helen, 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.

Yannakoudakis, Helen, 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.

Yasunaga, Michihiro, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763.

Yasunaga, Michihiro and Percy Liang. 2021. Break-it-fix-it: Unsupervised learning for program repair. In *International Conference on Machine Learning (ICML)*.

Yoshimoto, Ippei, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. 2013. NAIST at 2013 CoNLL grammatical error correction shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 26–33.

Yoshimura, Ryoma, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522.

Yu, Liang-Chih, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning chinese as a foreign language. In *Proceedings of the 22nd International Conference on Computers in Education*.

Yuan, Xun, Derek Pham, Sam Davidson, and Zhou Yu. 2022. ErAConD: Error annotated conversational dialog dataset for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 76–84.

Yuan, Zheng. 2017. *Grammatical error correction in non-native English*. Ph.D. thesis, University of Cambridge.

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

Yuan, Zheng, Ted Briscoe, and Mariano Felice. 2016. Candidate re-ranking for SMT-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 256–266.

Yuan, Zheng and Christopher Bryant. 2021. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84.

Yuan, Zheng and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61.

Yuan, Zheng, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239.

Yuan, Zheng, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of

two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736.

Zaghouani, Wajdi, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for non-native Arabic texts: Guidelines and corpus. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 129–139.

Zaghouani, Wajdi, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Zhang, Baolin. 2009. Features and functions of the HSK dynamic composition corpus (in Chinese). In *International Chinese Language Education*, pages 71–79.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: Open Pre-trained Transformer Language Models. *arXiv*, 2205.01068.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhang, Yue, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022b. MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction. In *Proceedings of the 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Zhang, Yue, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022c. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531.

Zhao, Wei, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165.

Zhao, Yuanyuan, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 439–445.

Zhao, Zewei and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1226–1233.

Zhou, Wangchunshu, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. Improving grammatical error correction with machine translation pairs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328.