Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction

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

Abstract

Until now, error type performance for Grammatical Error Correction (GEC) systems could only be measured in terms of recall because system output is not annotated. To overcome this problem, we introduce ERRANT, a grammatical ERRor ANnotation Toolkit designed to automatically extract edits from parallel original and corrected sentences and classify them according to a new, dataset-agnostic, rule-based framework. This not only facilitates error type evaluation at different levels of granularity, but can also be used to reduce annotator workload and standardise existing GEC datasets. Human experts rated the automatic edits as "Good" or "Acceptable" in at least 95% of cases, so we applied ERRANT to the system output of the CoNLL-2014 shared task to carry out a detailed error type analysis for the first time.

1 Introduction

Grammatical Error Correction (GEC) systems are often only evaluated in terms of overall performance because system hypotheses are not annotated. This can be misleading however, and a system that performs poorly overall may in fact outperform others at specific error types. This is significant because a robust *specialised* system is actually more desirable than a mediocre *general* system. Without an error type analysis however, this information is completely unknown.

The main aim of this paper is hence to rectify this situation and provide a method by which parallel error correction data can be automatically annotated with error type information. This not only facilitates error type evaluation, but can also be used to provide detailed error type feedback to non-native learners. Given that different corpora are also annotated according to different standards, we also attempted to standardise existing datasets under a common error type framework.

Our approach consists of two main steps. First, we automatically extract the edits between parallel original and corrected sentences by means of a linguistically-enhanced alignment algorithm (Felice et al., 2016) and second, we classify them according to a new, rule-based framework that relies solely on dataset-agnostic information such as lemma and part-of-speech. We demonstrate the value of our approach, which we call the ERRor ANnotation Toolkit (ERRANT)[1], by carrying out a detailed error type analysis of each system in the CoNLL-2014 shared task on grammatical error correction (Ng et al., 2014).

It is worth mentioning that despite an increased interest in GEC evaluation in recent years (Dahlmeier and Ng, 2012; Felice and Briscoe, 2015; Bryant and Ng, 2015; Napoles et al., 2015; Grundkiewicz et al., 2015; Sakaguchi et al., 2016), ERRANT is the only toolkit currently capable of producing error types scores.

2 Edit Extraction

The first stage of automatic annotation is *edit extraction*. Specifically, given an original and corrected sentence pair, we need to determine the start and end boundaries of any edits. This is fundamentally an alignment problem:

We	took	a	**guide**	tour	**on**		**center**	**city**	.
We	took	a	**guided**	tour	**of**	**the**	**city**	**center**	.

Table 1: A sample alignment between an original and corrected sentence (Felice et al., 2016).

[1] https://github.com/chrisjbryant/errant

The first attempt at automatic edit extraction was made by Swanson and Yamangil (2012), who simply used the Levenshtein distance to align parallel original and corrected sentences. As the Levenshtein distance only aligns individual tokens however, they also merged all adjacent non-matches in an effort to capture multi-token edits. Xue and Hwa (2014) subsequently improved on Swanson and Yamangil's work by training a maximum entropy classifier to predict whether edits should be merged or not.

Most recently, Felice et al. (2016) proposed a new method of edit extraction using a linguistically-enhanced alignment algorithm supported by a set of merging rules. More specifically, they incorporated various linguistic information, such as part-of-speech and lemma, into the cost function of the Damerau-Levenshtein[2] algorithm to make it more likely that tokens with similar linguistic properties aligned. This approach ultimately proved most effective at approximating human edits in several datasets (80-85% F_1), and so we use it in the present study.

3 Automatic Error Typing

Having extracted the edits, the next step is to assign them error types. While Swanson and Yamangil (2012) did this by means of maximum entropy classifiers, one disadvantage of this approach is that such classifiers are biased towards their particular training corpora. For example, a classifier trained on the First Certificate in English (FCE) corpus (Yannakoudakis et al., 2011) is unlikely to perform as well on the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier and Ng, 2012) or vice versa, because both corpora have been annotated according to different standards (cf. Xue and Hwa (2014)). Instead, a dataset-agnostic error type classifier is much more desirable.

3.1 A Rule-Based Error Type Framework

To solve this problem, we took inspiration from Swanson and Yamangil's (2012) observation that most error types are based on part-of-speech (POS) categories, and wrote a rule to classify an edit based only on its automatic POS tags. We then added another rule to similarly differentiate between Missing, Unnecessary and Replacement errors depending on whether tokens were inserted, deleted or substituted. Finally, we extended our approach to classify errors that are not well-characterised by POS, such as Spelling or Word Order, and ultimately assigned all error types based solely on automatically-obtained, objective properties of the data.

[2]Damerau-Levenshtein is an extension of Levenshtein that also handles transpositions; e.g. AB→BA

In total, we wrote roughly 50 rules. While many of them are very straightforward, significant attention was paid to discriminating between different kinds of verb errors. For example, despite all having the same correction, the following sentences contain different types of common learner errors:

(a)	He IS asleep now.	[*IS* → *is*]: orthography
(b)	He iss asleep now.	[*iss* → *is*]: spelling
(c)	He has asleep now.	[*has* → *is*]: verb
(d)	He being asleep now.	[*being* → *is*]: form
(e)	He was asleep now.	[*was* → *is*]: tense
(f)	He are asleep now.	[*are* → *is*]: SVA

To handle these cases, we hence wrote the following ordered rules:

1. Are the lower case forms of both sides of the edit the same? (a)
2. Is the original token a real word? (b)
3. Do both sides of the edit have the same lemma? (c)
4. Is one side of the edit a gerund (VBG) or participle (VBN)? (d)
5. Is one side of the edit in the past tense (VBD)? (e)
6. Is one side of the edit in the 3rd person present tense (VBZ)? (f)

While the final three rules could certainly be reordered, we informally found the above sequence performed best during development. It is also worth mentioning that this is a somewhat simplified example and that there are additional rules to discriminate between auxiliary verbs, main verbs and multi verb expressions. Nevertheless, the above case exemplifies our approach, and a more complete description of all rules is provided with the software.

Code	**Meaning**	**Description / Example**
ADJ	Adjective	*big → wide*
ADJ:FORM	Adjective Form	Comparative or superlative adjective errors. *goodest → best, bigger → biggest, more easy → easier*
ADV	Adverb	*speedily → quickly*
CONJ	Conjunction	*and → but*
CONTR	Contraction	*n't → not*
DET	Determiner	*the → a*
MORPH	Morphology	Tokens have the same lemma but nothing else in common. *quick (adj) → quickly (adv)*
NOUN	Noun	*person → people*
NOUN:INFL	Noun Inflection	Count-mass noun errors. *informations → information*
NOUN:NUM	Noun Number	*cat → cats*
NOUN:POSS	Noun Possessive	*friends → friend's*
ORTH	Orthography	Case and/or whitespace errors. *Bestfriend → best friend*
OTHER	Other	Errors that do not fall into any other category (e.g. paraphrasing). *at his best → well, job → professional*
PART	Particle	*(look) in → (look) at*
PREP	Preposition	*of → at*
PRON	Pronoun	*ours → ourselves*
PUNCT	Punctuation	*! → .*
SPELL	Spelling	*genectic → genetic, color → colour*
UNK	Unknown	The annotator detected an error but was unable to correct it.
VERB	Verb	*ambulate → walk*
VERB:FORM	Verb Form	Infinitives (with or without "to"), gerunds (-ing) and participles. *to eat → eating, dancing → danced*
VERB:INFL	Verb Inflection	Misapplication of tense morphology. *getted → got, fliped → flipped*
VERB:SVA	Subject-Verb Agreement	*(He) have → (He) has*
VERB:TENSE	Verb Tense	Includes inflectional and periphrastic tense, modal verbs and passivization. *eats → ate, eats → has eaten, eats → can eat, eats → was eaten*
WO	Word Order	*only can → can only*

Table 2: The list of 25 main error categories in our new framework with examples and explanations.

3.2 A Dataset-Agnostic Classifier

One of the key strengths of a rule-based approach is that by being dependent only on automatic mark-up information, our classifier is entirely dataset independent and does not require labelled training data. This is in contrast with machine learning approaches which not only learn dataset specific biases, but also presuppose the existence of sufficient quantities of training data.

A second significant advantage of our approach is that it is also always possible to determine precisely why an edit was assigned a particular error category. In contrast, human and machine learning classification decisions are often much less transparent.

Finally, by being fully deterministic, our approach bypasses bias effects altogether and should hence be more consistent.

3.3 Automatic Markup

The prerequisites for our rule-based classifier are that each token in both the original and corrected sentence is POS tagged, lemmatized, stemmed and dependency parsed. We use spaCy[3] v1.7.3 for all but the stemming, which is performed by the Lancaster Stemmer in NLTK.[4] Since fine-grained POS tags are often too detailed for the purposes of error evaluation, we also map spaCy's Penn Treebank style tags to the coarser set of Universal Dependency tags.[5] We use the latest Hunspell GB-large word list[6] to help classify non-word errors. The marked-up tokens in an edit span are then input to the classifier and an error type is returned.

3.4 Error Categories

The complete list of 25 error types in our new framework is shown in Table 2. Note that most of them can be prefixed with 'M:', 'R:' or 'U:', depending on whether they describe a Missing, Replacement, or Unnecessary edit, to enable

[3]https://spacy.io/
[4]http://www.nltk.org/
[5]http://universaldependencies.org/tagset-conversion/en-penn-uposf.html
[6]https://sourceforge.net/projects/wordlist/files/speller/2017.01.22/

evaluation at different levels of granularity (see Appendix A for all valid combinations). This means we can choose to evaluate, for example, only *replacement* errors (anything prefixed by 'R:'), only *noun* errors (anything suffixed with 'NOUN') or only *replacement noun* errors ('R:NOUN'). This flexibility allows us to make more detailed observations about different aspects of system performance.

One caveat concerning error scheme design is that it is always possible to add new categories for increasingly detailed error types; for instance, we currently label [*could* $\rightarrow$ *should*] a tense error, when it might otherwise be considered a modal error. The reason we do not call it a modal error, however, is because it would then become less clear how to handle other cases such as [*can* $\rightarrow$ *should*] and [*has eaten* $\rightarrow$ *should eat*], which might be considered a more complex combination of modal and tense error. As it is impractical to create new categories and rules to differentiate between such narrow distinctions however, our final framework aims to be a compromise between informativeness and practicality.

3.5 Classifier Evaluation

As our new error scheme is based solely on automatically obtained properties of the data, there are no gold standard labels against which to evaluate classifier performance. For this reason, we instead carried out a small-scale manual evaluation, where we simply asked 5 GEC researchers to rate the appropriateness of the predicted error types for 200 randomly chosen edits in context (100 from FCE-test and 100 from CoNLL-2014) as "Good", "Acceptable" or "Bad". "Good' meant the chosen type was the most appropriate for the given edit, "Acceptable" meant the chosen type was appropriate, but probably not optimum, while "Bad" meant the chosen type was not appropriate for the edit. Raters were warned that the edit boundaries had been determined automatically and hence might be unusual, but that they should focus on the appropriateness of the error type regardless of whether they agreed with the boundary or not.

It is worth stating that the main purpose of this evaluation was not to evaluate the specific strengths and weaknesses of the classifier, but rather ascertain how well humans believed the predicted error types characterised each edit. GEC is known to be a highly subjective task (Bryant and Ng, 2015) and so we were more interested in overall judgements than specific disagreements.

Rater	Good	Acceptable	Bad
1	92.0%	4.0%	4.0%
2	89.5%	6.5%	4.0%
3	83.0%	13.0%	4.0%
4	84.5%	11.0%	4.5%
5	82.5%	15.5%	2.0%
OVERALL	**86.3%**	**10.0%**	**3.7%**

Table 3: The percent distribution for how each expert rated the appropriateness of the predicted error types. E.g. Rater 3 considered 83% of all predicted types to be "Good".

The results from this evaluation are shown in Table 3. Significantly, all 5 raters considered at least 95% of the predicted error types to be either "Good" or "Acceptable", despite the degree of noise introduced by automatic edit extraction. Furthermore, whenever raters judged an edit as "Bad", this could usually be traced back to a POS or parse error; e.g. [*ring* $\rightarrow$ *rings*] might be considered a NOUN:NUM or VERB:SVA error depending on whether the POS tagger considered both sides of the edit nouns or verbs. Inter-annotator agreement was also good at 0.724 κ_{free} (Randolph, 2005).

In contrast, although incomparable on account of the different metric and error scheme, the best results using machine learning were between 50-70% F_1 (Felice et al., 2016). Ultimately however, we believe the high scores awarded by the raters validates the efficacy of our rule-based approach.

4 Error Type Scoring

Having described how to automatically annotate parallel sentences with ERRANT, we now also have a method to annotate system hypotheses; this is the first step towards an error type evaluation. Since no scorer is currently capable of calculating error type performance however (Dahlmeier and Ng, 2012; Felice and Briscoe, 2015; Napoles et al., 2015), we instead built our own.

Fortunately, one benefit of explicitly annotating system hypotheses is that it makes evaluation much more straightforward. In particular, for each sentence, we only need to compare the edits in the hypothesis against the edits in each respective reference and measure the overlap. Any edit with the same span and correction in both files is hence a true positive (TP), while unmatched edits in the hypothesis and references are false positives (FP) and false negatives (FN) respectively. These results can then be grouped by error type for the purposes of error type evaluation.

Finally, it is worth noting that this scorer is much simpler than other scorers in GEC which typically incorporate edit extraction or alignment directly into their algorithms. Our approach, on the other hand, treats edit extraction and evaluation as separate tasks.

4.1 Gold Reference vs. Auto Reference

Before evaluating an automatically annotated hypothesis against its reference, we must also address another mismatch: namely that hypothesis edits must be extracted and classified automatically, while reference edits are typically extracted and classified manually using a different framework. Since evaluation is now reduced to a straightforward comparison between two files however, it is especially important that the hypothesis and references are both processed in the same way. For instance, a hypothesis edit [*have eating* $\rightarrow$ *has eaten*] will not match the reference edits [*have* $\rightarrow$ *has*] and [*eating* $\rightarrow$ *eaten*] because the former is one edit while the latter is two edits, even though they equate to the same thing.

To solve this problem, we can reprocess the references in the same way as the hypotheses. In other words, we can apply ERRANT to the references such that each reference edit is subject to the same automatic extraction and classification criteria as each hypothesis edit. While it may seem unorthodox to discard gold reference information in favour of automatic reference information, this is necessary to minimise the difference between hypothesis and reference edits and also standardise error type annotations.

To show that automatic references are feasible alternatives to gold references, we evaluated each team in the CoNLL-2014 shared task using both types of reference with the M^2 scorer (Dahlmeier and Ng, 2012), the *de facto* standard of GEC evaluation, and our own scorer. Table 4 hence shows that there is little difference between the overall scores for each team, and we formally validated this hypothesis for precision, recall and F$_{0.5}$ by means of bootstrap significance testing (Efron and Tibshirani, 1993). Ultimately, we found no statistically significant difference between automatic and gold references (1,000 iterations, $p > .05$) which leads us to conclude that our automatic references are qualitatively as good as human references.

	M^2 Scorer		**Our Scorer**	
Team	**Gold**	**Auto**	**Gold**	**Auto**
AMU	35.01	35.05	31.95	32.25
CAMB	37.33	37.34	33.39	34.01
CUUI	36.79	37.59	33.32	34.64
IITB	5.90	5.96	5.67	5.74
IPN	7.09	7.68	5.86	6.14
NTHU	29.92	29.77	25.62	25.66
PKU	25.32	25.38	23.40	23.60
POST	30.88	31.01	27.54	27.99
RAC	26.68	26.88	22.83	23.15
SJTU	15.19	15.22	14.85	14.89
UFC	7.84	7.89	7.84	7.89
UMC	25.37	25.45	23.08	23.52

Table 4: Overall scores for each team in CoNLL-2014 using gold and auto references with both the M^2 scorer and our simpler edit comparison approach. All scores are in terms of F$_{0.5}$.

4.2 Comparison with the M^2 Scorer

Despite using the same metric, Table 4 also shows that the M^2 scorer tends to produce slightly higher F$_{0.5}$ scores than our own. This initially led us to believe that our scorer was underestimating performance, but we subsequently found that instead the M^2 scorer tends to overestimate performance (cf. Felice and Briscoe (2015) and Napoles et al. (2015)).

In particular, given a choice between matching [*have eating* $\rightarrow$ *has eaten*] from Annotator 1 or [*have* $\rightarrow$ *has*] and [*eating* $\rightarrow$ *eaten*] from Annotator 2, the M^2 scorer will always choose Annotator 2 because two true positives (TP) are worth more than one. Similarly, whenever the scorer encounters two false positives (FP) within a certain distance of each other,[7] it merges them and treats them as one false positive; e.g. [*is a cat* $\rightarrow$ *are a cats*] is selected over [*is* $\rightarrow$ *are*] and [*cat* $\rightarrow$ *cats*] even though these edits are best handled separately. In other words, the M^2 scorer exploits its dynamic edit boundary prediction to artificially maximise true positives and minimise false positives and hence produce slightly inflated scores.

[7]The distance is controlled by the *max_unchanged_words* parameter which is set to 2 by default.

	AMU			CAMB			CUUI			IITB		
Type	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Missing	43.94	14.32	31.08	45.96	29.71	41.43	26.37	18.16	24.18	15.38	0.59	2.56
Replacement	37.22	26.92	34.57	37.53	28.12	35.18	45.90	22.98	38.27	29.85	1.49	6.22
Unnecessary	-	-	-	25.51	27.47	25.88	34.20	33.33	34.02	46.15	1.53	6.77

	IPN			NTHU			PKU			POST		
Type	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Missing	2.86	0.29	1.04	34.33	11.39	24.47	33.33	4.37	14.34	31.14	13.13	24.44
Replacement	9.87	3.86	7.53	27.61	19.15	25.37	29.62	18.33	26.37	33.16	19.33	29.01
Unnecessary	0.00	0.00	0.00	34.76	15.97	28.14	0.00	0.00	0.00	26.32	32.84	27.40

	RAC			SJTU			UFC			UMC		
Type	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Missing	1.52	0.27	0.79	62.50	4.44	17.28	-	-	-	40.08	23.57	35.16
Replacement	29.41	20.82	27.17	50.54	3.43	13.47	72.00	2.64	11.52	34.71	9.70	22.90
Unnecessary	0.00	0.00	0.00	17.65	11.36	15.89	-	-	-	16.86	17.17	16.92

Table 5: Precision, recall and $F_{0.5}$ for Missing, Unnecessary, and Replacement errors for each team. A dash indicates the team's system did not attempt to correct the given error type (TP+FP = 0).

5 CoNLL-2014 Shared Task Analysis

To demonstrate the value of ERRANT, we applied it to the data produced in the CoNLL-2014 shared task (Ng et al., 2014). Specifically, we automatically annotated all the system hypotheses and official reference files.[8] Although ERRANT can be applied to any dataset of parallel sentences, we chose to evaluate on CoNLL-2014 because it represents the largest collection of publicly available GEC system output. For more information about the systems in CoNLL-2014, we refer the reader to the shared task paper.

5.1 Edit Operation

In our first category experiment, we simply investigated the performance of each system in terms of Missing, Replacement and Unnecessary edits. The results are shown in Table 5 with additional information in Appendix B, Table 10.

The most surprising result is that five teams (AMU, IPN, PKU, RAC, UFC) failed to correct any unnecessary token errors at all. This is noteworthy because unnecessary token errors account for roughly 25% of all errors in the CoNLL-2014 test data and so failing to address them significantly limits a system's maximum performance. While the reason for this is clear in some cases, e.g. UFC's rule-based system was never designed to tackle unnecessary tokens (Gupta, 2014), it is less clear in others, e.g. there is no obvious reason why AMU's SMT system failed to learn when to delete tokens (Junczys-Dowmunt and Grundkiewicz, 2014). AMU's result is especially remarkable given that their system still came 3rd overall despite this limitation.

In contrast, CUUI's classifier approach (Rozovskaya et al., 2014) was the most successful at correcting not only unnecessary token errors, but also replacement token errors, while CAMB's hybrid MT approach (Felice et al., 2014) significantly outperformed all others in terms of missing token errors. It would hence make sense to combine these two approaches, and indeed recent research has shown this improves overall performance (Rozovskaya and Roth, 2016).

5.2 General Error Types

Table 6 shows precision, recall and $F_{0.5}$ for each of the error types in our proposed framework for each team in CoNLL-2014. As some error types are more common than others, we also provide the TP, FP and FN counts used to make this table in Appendix B, Table 11.

Overall, CAMB was the most successful team in terms of error types, achieving the highest F-score in 10 (out of 24) error categories, followed by AMU, who scored highest in 6 categories. All but 3 teams (IITB, IPN and POST) achieved the best score in at least 1 category, which suggests that different approaches to GEC complement different error types. Only CAMB attempted to correct at least 1 error from every category.

Other interesting observations we can make from this table include:

[8] http://www.comp.nus.edu.sg/~nlp/conll14st.html

		AMU	CAMB	CUUI	IITB	IPN	NTHU	PKU	POST	RAC	SJTU	UFC	UMC
ADJ	**P**	4.88	9.09	-	0.00	0.00	0.00	66.67	0.00	12.50	0.00	-	0.00
	R	6.67	13.89	-	0.00	0.00	0.00	7.14	0.00	3.57	0.00	-	0.00
	$F_{0.5}$	5.15	9.77	-	0.00	0.00	0.00	**25.00**	0.00	8.33	0.00	-	0.00
ADJ:FORM	**P**	55.56	75.00	100.00	100.00	0.00	33.33	100.00	50.00	8.00	-	-	100.00
	R	62.50	60.00	33.33	40.00	0.00	37.50	28.57	14.29	40.00	-	-	60.00
	$F_{0.5}$	56.82	71.43	71.43	76.92	0.00	34.09	66.67	33.33	9.52	-	-	**88.24**
ADV	**P**	6.67	11.54	0.00	0.00	0.00	0.00	0.00	-	0.00	4.76	-	8.77
	R	2.94	20.45	0.00	0.00	0.00	0.00	0.00	-	0.00	3.03	-	12.50
	$F_{0.5}$	5.32	**12.64**	0.00	0.00	0.00	0.00	0.00	-	0.00	4.27	-	9.33
CONJ	**P**	6.25	0.00	-	-	0.00	0.00	-	-	-	0.00	-	0.00
	R	7.69	0.00	-	-	0.00	0.00	-	-	-	0.00	-	0.00
	$F_{0.5}$	**6.49**	0.00	-	-	0.00	0.00	-	-	-	0.00	-	0.00
CONTR	**P**	29.17	40.00	46.15	-	0.00	-	-	33.33	0.00	66.67	-	28.57
	R	100.00	33.33	85.71	-	0.00	-	-	57.14	0.00	40.00	-	33.33
	$F_{0.5}$	33.98	38.46	50.85	-	0.00	-	-	36.36	0.00	**58.82**	-	29.41
DET	**P**	33.33	36.16	30.92	21.43	0.00	36.03	29.35	26.09	0.00	43.88	-	36.21
	R	14.09	43.03	51.91	0.92	0.00	28.46	7.85	49.41	0.00	12.54	-	23.66
	$F_{0.5}$	26.18	**37.35**	33.64	3.92	0.00	34.21	18.96	28.81	0.00	29.25	-	32.74
MORPH	**P**	55.56	59.15	55.88	28.57	1.16	27.87	20.80	27.78	32.69	100.00	40.00	43.75
	R	48.91	47.73	20.88	5.41	1.39	21.52	30.59	12.50	21.25	2.74	5.00	15.91
	$F_{0.5}$	54.09	**56.45**	41.85	15.38	1.20	26.32	22.22	22.32	29.51	12.35	16.67	32.41
NOUN	**P**	20.90	25.27	0.00	28.57	4.35	0.00	0.00	10.00	10.53	0.00	-	27.78
	R	12.39	19.49	0.00	2.20	2.17	0.00	0.00	1.92	1.92	0.00	-	9.90
	$F_{0.5}$	18.37	**23.86**	0.00	8.40	3.62	0.00	0.00	5.43	5.56	0.00	-	20.41
NOUN:INFL	**P**	60.00	60.00	50.00	-	25.00	100.00	62.50	66.67	66.67	0.00	-	-
	R	85.71	66.67	71.43	-	16.67	33.33	62.50	57.14	66.67	0.00	-	-
	$F_{0.5}$	63.83	61.22	53.19	-	22.73	**71.43**	62.50	64.52	66.67	0.00	-	-
NOUN:NUM	**P**	49.42	44.20	44.06	41.18	14.38	44.05	29.39	31.05	29.00	54.29	-	44.29
	R	56.14	53.74	59.49	3.87	11.28	47.62	42.54	56.20	36.45	10.27	-	16.94
	$F_{0.5}$	**50.63**	45.83	46.47	14.06	13.63	44.72	31.33	34.10	30.23	29.23	-	33.48
NOUN:POSS	**P**	20.00	66.67	-	-	-	-	14.29	0.00	0.00	25.00	-	50.00
	R	14.29	10.53	-	-	-	-	5.26	0.00	0.00	4.55	-	5.00
	$F_{0.5}$	18.52	**32.26**	-	-	-	-	10.64	0.00	0.00	13.16	-	17.86
ORTH	**P**	60.00	66.67	73.81	-	3.45	0.00	28.57	49.32	16.57	-	-	50.00
	R	11.11	40.00	59.62	-	4.55	0.00	6.90	64.29	49.12	-	-	17.24
	$F_{0.5}$	31.91	58.82	**70.45**	-	3.62	0.00	17.54	51.72	19.10	-	-	36.23
OTHER	**P**	20.34	23.60	10.34	0.00	2.33	1.37	14.29	10.00	0.00	0.00	-	11.58
	R	6.92	10.03	0.83	0.00	0.31	0.58	0.58	1.13	0.00	0.00	-	3.15
	$F_{0.5}$	14.65	**18.57**	3.14	0.00	1.01	1.07	2.49	3.90	0.00	0.00	-	7.54
PART	**P**	71.43	33.33	25.00	-	-	16.67	-	-	-	50.00	-	20.00
	R	20.83	15.38	4.76	-	-	21.74	-	-	-	9.52	-	11.11
	$F_{0.5}$	**48.08**	27.03	13.51	-	-	17.48	-	-	-	27.03	-	17.24
PREP	**P**	47.56	41.44	33.33	75.00	0.00	10.71	-	21.74	0.00	36.59	-	20.53
	R	16.05	35.66	13.49	1.44	0.00	12.35	-	2.17	0.00	7.18	-	13.36
	$F_{0.5}$	34.15	**40.14**	25.76	6.70	0.00	11.01	-	7.76	0.00	20.11	-	18.54
PRON	**P**	41.18	20.37	0.00	0.00	11.11	50.00	100.00	27.27	5.00	0.00	-	22.92
	R	9.72	13.41	0.00	0.00	1.69	2.82	1.54	4.62	1.52	0.00	-	13.92
	$F_{0.5}$	**25.00**	18.46	0.00	0.00	5.26	11.49	7.25	13.76	3.42	0.00	-	20.30
PUNCT	**P**	25.00	60.47	37.21	100.00	0.00	44.83	-	27.27	0.00	5.00	-	43.02
	R	3.52	15.48	10.60	1.85	0.00	8.97	-	6.34	0.00	0.96	-	23.13
	$F_{0.5}$	11.26	**38.24**	24.77	8.62	0.00	24.90	-	16.42	0.00	2.72	-	36.71
SPELL	**P**	76.92	77.55	0.00	0.00	25.00	0.00	44.17	68.63	73.98	-	-	100.00
	R	63.83	41.76	0.00	0.00	4.23	0.00	71.29	71.43	85.85	-	-	1.37
	$F_{0.5}$	73.89	66.20	0.00	0.00	12.61	0.00	47.81	69.17	**76.09**	-	-	6.49
VERB	**P**	18.84	15.12	-	0.00	7.69	0.00	14.29	0.00	0.00	0.00	-	16.33
	R	8.23	8.33	-	0.00	0.74	0.00	0.70	0.00	0.00	0.00	-	5.37
	$F_{0.5}$	**14.98**	13.00	-	0.00	2.66	0.00	2.94	0.00	0.00	0.00	-	11.59
VERB:FORM	**P**	34.92	36.36	68.75	0.00	8.77	35.11	30.77	25.00	34.41	28.57	-	31.11
	R	23.40	25.00	24.18	0.00	5.75	35.11	35.56	3.45	32.65	4.65	-	16.09
	$F_{0.5}$	31.79	33.33	**50.23**	0.00	7.94	35.11	31.62	11.11	34.04	14.08	-	26.22
VERB:INFL	**P**	100.00	100.00	-	-	100.00	100.00	50.00	100.00	100.00	-	0.00	-
	R	100.00	100.00	-	-	50.00	50.00	50.00	50.00	100.00	-	0.00	-
	$F_{0.5}$	**100.00**	**100.00**	-	-	83.33	83.33	50.00	83.33	**100.00**	-	0.00	-
VERB:SVA	**P**	49.09	44.05	54.80	50.00	24.56	50.56	56.25	32.69	35.56	59.09	81.58	60.00
	R	27.55	32.74	71.85	1.12	14.58	67.16	18.75	17.35	31.07	13.83	29.25	15.00
	$F_{0.5}$	42.45	41.20	57.53	5.15	21.60	53.19	40.18	27.78	34.56	35.71	**60.08**	37.50
VERB:TENSE	**P**	20.55	26.27	70.00	66.67	3.70	31.25	9.38	20.00	22.78	14.81	100.00	31.25
	R	8.72	17.51	4.12	1.25	0.61	2.98	3.66	2.31	20.57	2.45	0.63	12.05
	$F_{0.5}$	16.16	**23.88**	16.67	5.81	1.84	10.78	7.14	7.91	22.30	7.38	3.05	23.70
WO	**P**	-	38.89	0.00	66.67	-	-	-	0.00	0.00	-	-	41.18
	R	-	33.33	0.00	14.29	-	-	-	0.00	0.00	-	-	35.00
	$F_{0.5}$	-	37.63	0.00	38.46	-	-	-	0.00	0.00	-	-	**39.77**

Table 6: Precision, recall and $F_{0.5}$ for each team and error type. A dash indicates the team's system did not attempt to correct the given error type (TP+FP = 0). The highest F-score for each type is highlighted.

	CAMB		
Type	**P**	**R**	**$F_{0.5}$**
M:DET	43.20	51.77	44.68
R:DET	19.33	35.37	21.26
U:DET	43.75	39.90	42.92
DET	36.16	43.03	37.35

	CUUI		
Type	**P**	**R**	**$F_{0.5}$**
M:DET	23.86	45.00	26.34
R:DET	27.03	24.39	26.46
U:DET	36.19	66.37	39.81
DET	30.92	51.91	33.64

Table 7: Detailed breakdown of Determiner errors for two teams.

- Despite the prevalence of spell checkers nowadays, many teams did not seem to employ them; this would have been an easy way to boost overall performance.
- Although several teams built specialised classifiers for DET and PREP errors, CAMB's hybrid MT approach still outperformed them. This might be because the classifiers were trained using a different error type framework however.
- CUUI's classifiers significantly outperformed all other approaches at ORTH and VERB:FORM errors. This suggests classifiers are well-suited to these error types.
- Although UFC's rule-based approach was the best at VERB:SVA errors, CUUI's classifier was not very far behind.
- Only AMU managed to correct any CONJ errors.
- Content word errors (i.e. ADJ, ADV, NOUN and VERB) were unsurprisingly very difficult for all teams.

5.3 Detailed Error Types

In addition to analysing general error types, the modular design of our framework also allows us to evaluate error type performance at an even greater level of detail. For example, Table 7 shows the breakdown of Determiner errors for two teams using different approaches in terms of edit operation. Note that this is a representative example of detailed error type performance, as an analysis of all error type combinations for all teams would take up too much space.

Team	P	R	$F_{0.5}$
AMU	16.90	5.33	11.79
CAMB	27.22	17.06	24.32
CUUI	15.69	3.67	9.48
IITB	28.57	0.94	4.15
IPN	3.33	0.47	1.51
NTHU	0.00	0.00	0.00
PKU	25.00	1.40	5.73
POST	12.77	2.82	7.48
RAC	2.96	2.82	2.93
SJTU	10.00	0.47	1.99
UFC	-	-	-
UMC	19.82	9.82	16.47

Table 8: Each team's performance at correcting multi-token edits; i.e. there are at least two tokens on one side of the edit.

While CAMB's hybrid MT approach achieved a higher score than CUUI's classifier overall, our more detailed evaluation reveals that CUUI actually outperformed CAMB at Replacement Determiner errors. We also learn that CAMB scored twice as highly on M:DET and U:DET than it did on R:DET and that CUUI's significantly higher U:DET recall was offset by a lower precision. Ultimately, this shows that even though one approach might be better than another overall, different approaches may still have complementary strengths.

5.4 Multi Token Errors

Another benefit of explicitly annotating all hypothesis edits is that edit spans become fixed; this means we can evaluate system performance in terms of edit size. Table 8 hence shows the overall performance for each team at correcting multi-token edits, where a multi-token edit is an edit that has at least two tokens on either side. In the CoNLL-2014 test set, there are roughly 220 such edits (about 10% of all edits).

In general, teams did not do well at multi-token edits. In fact only three teams achieved scores greater than 10% $F_{0.5}$ and all of them used MT (AMU, CAMB, UMC). This is significant because recent work has suggested that the main goal of GEC should be to produce fluent-sounding, rather than just grammatical sentences, even though this often requires complex multi-token edits (Sakaguchi et al., 2016). If no system is particularly adept at correcting multi-token errors however, robust fluency correction will likely require more sophisticated methods than are currently available.

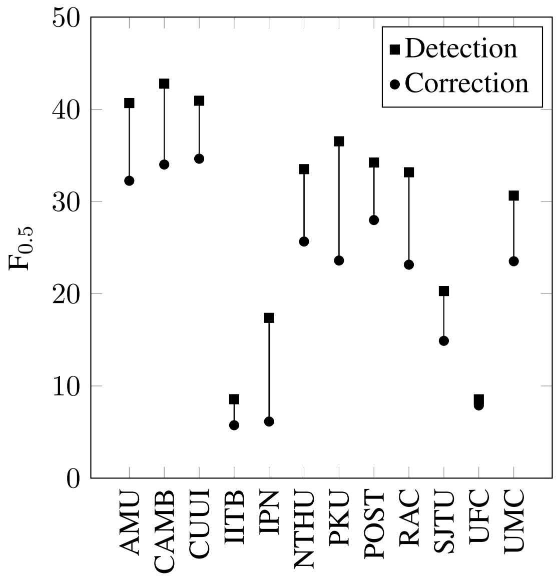

Figure 1: The difference between detection and correction scores for each team overall.

5.5 Detection vs. Correction

Another important aspect of GEC that is seldom reported in the literature is that of error detection; i.e. the extent to which a system can identify erroneous tokens in text. This can be calculated by comparing the edit overlap between the hypothesis and reference files *regardless of the proposed correction* in a manner similar to Recognition evaluation in the HOO shared tasks for GEC (Dale and Kilgarriff, 2011).

Figure 1 hence shows how each team's score for detection differed in relation to their score for correction. While CAMB scored highest for detection overall, it is interesting to note that CUUI ultimately performed slightly better than CAMB at correction. This suggests CUUI was more successful at correcting the errors they detected than CAMB. In contrast, IPN and PKU are notable for detecting significantly more errors than they were able to correct. Nevertheless, a system's ability to detect errors, even if it is unable to correct them, is still likely to be valuable information to a learner (Rei and Yannakoudakis, 2016).

Finally, although we do not do so here, our scorer is also capable of providing a detailed error type breakdown for detection.

6 Conclusion

In this paper, we described ERRANT, a grammatical ERRor ANnotation Toolkit designed to automatically annotate parallel error correction data with explicit edit spans and error type information. ERRANT can be used to not only facilitate a detailed error type evaluation in GEC, but also to standardise existing error correction corpora and reduce annotator workload. We release ERRANT with this paper.

Our approach makes use of previous work to align sentences based on linguistic intuition and then introduces a new rule-based framework to classify edits. This framework is entirely dataset independent, and relies only on automatically obtained information such as POS tags and lemmas. A small-scale evaluation of our classifier found that each rater considered >95% of the predicted error types as either "Good" (85%) or "Acceptable" (10%).

We demonstrated the value of ERRANT by carrying out a detailed evaluation of system error type performance for all teams in the CoNLL-2014 shared task on Grammatical Error Correction. We found that different systems had different strengths and weaknesses which we hope researchers can exploit to further improve general performance.

References

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 697–707. http://www.aclweb.org/anthology/P15-1068.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Montréal, Canada, pages 568–572. http://www.aclweb.org/anthology/N12-1067.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Association for Computational Linguistics, Stroudsburg, PA, USA, ENLG '11, pages 242–249. http://dl.acm.org/citation.cfm?id=2187681.2187725.

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

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 578–587. http://www.aclweb.org/anthology/N15-1060.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 825–835. http://aclweb.org/anthology/C16-1079.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 15–24. http://www.aclweb.org/anthology/W14-1702.

Roman Grundkiewicz, 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*. Association for Computational Linguistics, Lisbon, Portugal, pages 461–470. http://aclweb.org/anthology/D15-1052.

Anubhav Gupta. 2014. Grammatical error detection using tagger disagreement. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 49–52. http://www.aclweb.org/anthology/W14-1706.

Marcin Junczys-Dowmunt 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*. Association for Computational Linguistics, Baltimore, Maryland, pages 25–33. http://www.aclweb.org/anthology/W14-1703.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 588–593. http://www.aclweb.org/anthology/P15-2097.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. ACL, Baltimore, Maryland, USA, pages 1–14. http://aclweb.org/anthology/W/W14/W14-1701.pdf.

Justus J. Randolph. 2005. Free-marginal multirater kappa: An alternative to Fleiss' fixed-marginal multirater kappa. *Joensuu University Learning and Instruction Symposium* http://files.eric.ed.gov/fulltext/ED490661.pdf.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1181–1191. http://www.aclweb.org/anthology/P16-1112.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 34–42. http://www.aclweb.org/anthology/W14-1704.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany, pages 2205–2215. http://aclweb.org/anthology/P16-1208.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics* 4:169–182. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/800.

Ben Swanson and Elif Yamangil. 2012. Correction detection and error type selection as an ESL educational aid. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Montréal, Canada, pages 357–361. http://www.aclweb.org/anthology/N12-1037.

Huichao Xue and Rebecca Hwa. 2014. Improved correction detection in revised ESL sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 599–604. http://www.aclweb.org/anthology/P14-2098.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 180–189. http://www.aclweb.org/anthology/P11-1019.

A Complete list of valid error code combinations

			Operation Tier		
		Type	**Missing**	**Unnecessary**	**Replacement**
Token Tier	**Part Of Speech**	Adjective	M:ADJ	U:ADJ	R:ADJ
		Adverb	M:ADV	U:ADV	R:ADV
		Conjunction	M:CONJ	U:CONJ	R:CONJ
		Determiner	M:DET	U:DET	R:DET
		Noun	M:NOUN	U:NOUN	R:NOUN
		Particle	M:PART	U:PART	R:PART
		Preposition	M:PREP	U:PREP	R:PREP
		Pronoun	M:PRON	U:PRON	R:PRON
		Punctuation	M:PUNCT	U:PUNCT	R:PUNCT
		Verb	M:VERB	U:VERB	R:VERB
	Other	Contraction	M:CONTR	U:CONTR	R:CONTR
		Morphology	-	-	R:MORPH
		Orthography	-	-	R:ORTH
		Other	M:OTHER	U:OTHER	R:OTHER
		Spelling	-	-	R:SPELL
		Word Order	-	-	R:WO
Morphology Tier		Adjective Form	-	-	R:ADJ:FORM
		Noun Inflection	-	-	R:NOUN:INFL
		Noun Number	-	-	R:NOUN:NUM
		Noun Possessive	M:NOUN:POSS	U:NOUN:POSS	R:NOUN:POSS
		Verb Form	M:VERB:FORM	U:VERB:FORM	R:VERB:FORM
		Verb Inflection	-	-	R:VERB:INFL
		Verb Agreement	-	-	R:VERB:SVA
		Verb Tense	M:VERB:TENSE	U:VERB:TENSE	R:VERB:TENSE

Table 9: There are 55 total possible error types. This table shows all of them except UNK, which indicates an uncorrected error. A dash indicates an impossible combination.

B TP, FP and FN counts for various CoNLL-2014 results

	AMU			CAMB			CUUI			IITB		
Type	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**
Missing	58	74	347	131	154	310	77	215	347	2	11	336
Replacement	428	722	1162	477	794	1219	381	449	1277	20	47	1320
Unnecessary	0	0	412	125	365	330	158	304	316	6	7	385

	IPN			NTHU			PKU			POST		
Type	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**
Missing	1	34	339	46	88	358	16	32	350	52	115	344
Replacement	53	484	1319	299	784	1262	279	663	1243	312	629	1302
Unnecessary	0	2	389	65	122	342	0	1	397	155	434	317

	RAC			SJTU			UFC			UMC		
Type	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**
Missing	1	65	368	15	9	323	0	0	339	99	148	321
Replacement	325	780	1236	47	46	1325	36	14	1326	143	269	1331
Unnecessary	0	5	407	45	210	351	0	0	381	74	365	357

Table 10: True Positive, False Positive and False Negative counts for each team in terms of Missing, Replacement and Unnecessary edits. The total number of edits may vary for each system, as this depends on the individual references that are chosen during evaluation. These results were used to make Table 5.

		AMU	CAMB	CUUI	IITB	IPN	NTHU	PKU	POST	RAC	SJTU	UFC	UMC
ADJ	**TP**	2	5	0	0	0	0	2	0	1	0	0	0
	FP	39	50	0	3	2	3	1	4	7	8	0	21
	FN	28	31	30	23	23	25	26	33	27	20	20	26
ADJ:FORM	**TP**	5	6	3	2	0	3	2	1	2	0	0	3
	FP	4	2	0	0	1	6	0	1	23	0	0	0
	FN	3	4	6	3	5	5	5	6	3	5	5	2
ADV	**TP**	1	9	0	0	0	0	0	0	0	1	0	5
	FP	14	69	1	1	1	2	1	0	4	20	0	52
	FN	33	35	36	32	33	35	37	41	37	32	33	35
CONJ	**TP**	1	0	0	0	0	0	0	0	0	0	0	0
	FP	15	18	0	0	1	1	0	0	0	6	0	26
	FN	12	15	14	12	12	13	12	15	13	13	12	14
CONTR	**TP**	7	2	6	0	0	0	0	4	0	2	0	2
	FP	17	3	7	0	1	0	0	8	1	1	0	5
	FN	0	4	1	5	5	5	5	3	5	3	5	4
DET	**TP**	52	179	231	3	0	107	27	210	0	43	0	88
	FP	104	316	516	11	13	190	65	595	9	55	0	155
	FN	317	237	214	324	325	269	317	215	346	300	327	284
MORPH	**TP**	45	42	19	4	1	17	26	10	17	2	4	14
	FP	36	29	15	10	85	44	99	26	35	0	6	18
	FN	47	46	72	70	71	62	59	70	63	71	76	74
NOUN	**TP**	14	23	0	2	2	0	0	2	2	0	0	10
	FP	53	68	5	5	44	9	29	18	17	16	0	26
	FN	99	95	109	89	90	102	103	102	102	93	92	91
NOUN:INFL	**TP**	6	6	5	0	1	2	5	4	4	0	0	0
	FP	4	4	5	0	3	0	3	2	2	1	0	0
	FN	1	3	2	6	5	4	3	3	2	6	6	6
NOUN:NUM	**TP**	128	122	141	7	22	100	97	136	78	19	0	31
	FP	131	154	179	10	131	127	233	302	191	16	0	39
	FN	100	105	96	174	173	110	131	106	136	166	178	152
NOUN:POSS	**TP**	3	2	0	0	0	0	1	0	0	1	0	1
	FP	12	1	0	0	0	0	6	1	38	3	0	1
	FN	18	17	20	18	19	22	18	21	20	21	20	19
ORTH	**TP**	3	14	31	0	1	0	2	36	28	0	0	5
	FP	2	7	11	0	28	1	5	37	141	0	0	5
	FN	24	21	21	21	21	27	27	20	29	24	21	24
OTHER	**TP**	24	38	3	0	1	2	2	4	0	0	0	11
	FP	94	123	26	8	42	144	12	36	52	11	0	84
	FN	323	341	358	329	322	345	343	349	346	323	327	338
PART	**TP**	5	4	1	0	0	5	0	0	0	2	0	2
	FP	2	8	3	0	0	25	0	0	0	2	0	8
	FN	19	22	20	19	21	18	21	20	19	19	17	16
PREP	**TP**	39	92	34	3	0	30	0	5	0	15	0	31
	FP	43	130	68	1	2	250	0	18	3	26	0	120
	FN	204	166	218	205	207	213	219	225	215	194	206	201
PRON	**TP**	7	11	0	0	1	2	1	3	1	0	0	11
	FP	10	43	1	5	8	2	0	8	19	22	0	37
	FN	65	71	63	57	58	69	64	62	65	62	62	68
PUNCT	**TP**	5	26	16	2	0	13	0	9	0	1	0	37
	FP	15	17	27	0	16	16	0	24	29	19	0	49
	FN	137	142	135	106	114	132	123	133	129	103	109	123
SPELL	**TP**	60	38	0	0	3	0	72	70	91	0	0	1
	FP	18	11	1	1	9	2	91	32	32	0	0	0
	FN	34	53	74	68	68	74	29	28	15	70	70	72
VERB	**TP**	13	13	0	0	1	0	1	0	0	0	0	8
	FP	56	73	0	6	12	12	6	4	5	17	0	41
	FN	145	143	165	133	135	152	141	164	151	139	131	141
VERB:FORM	**TP**	22	24	22	0	5	33	32	3	32	4	0	14
	FP	41	42	10	1	52	61	72	9	61	10	0	31
	FN	72	72	69	87	82	61	58	84	66	82	82	73
VERB:INFL	**TP**	2	2	0	0	1	1	1	1	2	0	0	0
	FP	0	0	0	0	0	0	1	0	0	0	1	0
	FN	0	0	2	2	1	1	1	1	0	2	2	2
VERB:SVA	**TP**	27	37	97	1	14	90	18	17	32	13	31	15
	FP	28	47	80	1	43	88	14	35	58	9	7	10
	FN	71	76	38	88	82	44	78	81	71	81	75	85
VERB:TENSE	**TP**	15	31	7	2	1	5	6	4	36	4	1	20
	FP	58	87	3	1	26	11	58	16	122	23	0	44
	FN	157	146	163	158	163	163	158	169	139	159	159	146
WO	**TP**	0	7	0	2	0	0	0	0	0	0	0	7
	FP	0	11	10	1	0	0	0	2	1	0	0	10
	FN	12	14	14	12	12	11	12	12	12	11	11	13

Table 11: True Positive, False Positive and False Negative counts for each error type for each team. These results were used to make Table 6.