The Effect of Error Rate in Artificially Generated Data for Automatic Preposition and Determiner Correction

Fraser Bowen and **Jon Dehdari** and **Josef van Genabith**
University of Saarland
Deutsches Forschungsinstitut für Künstliche Intelligenz

Abstract

In this research we investigate the impact of mismatches in the density and type of error between training and test data on a neural system correcting preposition and determiner errors. We use synthetically produced training data to control error density and type, and "real" error data for testing. Our results show it is possible to combine error types, although prepositions and determiners behave differently in terms of how much error should be artificially introduced into the training data in order to get the best results.

1 Introduction

The field of Grammatical Error Correction (GEC) is currently dominated by neural translation models, specifically sequence-to-sequence translation. However, despite offering substantial improvements on the well-established statistical machine translation approach to GEC, neural networks come with their own challenges.

Firstly, neural models require a large amount of training data, however the amount of annotated learner English consisting of source (original text) and target (corrected text) is low. Models are at risk of overfitting, simply because the volume of data is not high enough. Secondly, the data that has been used up until now does not generalise very well across different test sets. This means that there has been some success in correcting errors, but only from test sets that are in some sense similar to the training data. Thirdly, it is generally unknown how erroneous the test data is, and if the training data has a different distribution of errors, it is likely that unwanted corrections will be made, or required corrections will be missed.

Currently, there is research into generating artificial data for training neural models, specifically data that resembles learner English (Cahill et al., 2013; Rozovskaya and Roth, 2010; Felice, 2016; Liu and Liu, 2016). The artificial data is generated from monolingual sentences of grammatical English by systematically introducing noise into it. This way, training data consisting of sentences with both "incorrect" and "correct" versions can be generated from monolingual data, which is easily accessible. There is also evidence that artificially generated data can generalise a GEC system better than simply using manually procured correction data (Cahill et al., 2013).

A third advantage of synthetically introducing noise into a corpus is the ability to control how much noise, and which noise, is introduced. The first main question of our research is how the amount of noise introduced into the corpus affects a neural model's behaviour at test time with respect to mismatches in error density and error type between training and test data. Artificial data lends itself to this kind of research, thanks to the control over the corpus.

Up until now, the effect of the amount of errors in the training corpus has only been explored with prepositions specifically (Cahill et al., 2013). We begin by extending this line of research to determiners. The second research question is then: how do two different types of error interact? It is quite possible that introducing many types of frequent grammatical errors one after the other would not create convincing artificial learner data, because several types of error can affect the same word, and a neural model may not be able to learn to combine them in this way.

Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 68–76
Copenhagen, Denmark, September 7, 2017.

2 Related Work

Currently the best results in GEC have used neural machine translation. Yuan and Briscoe (2016) achieved the best scores using a 2-layer encoder-decoder system with attention, trained on the Cambridge Learner Corpus (CLC), a large data set of two million correction Learner English sentences. The CLC is not publicly available, which has inspired the use of automatically generated data with neural models. Liu and Liu (2016) have done exactly this with 16 different types of errors. Their success, although small compared to using manually annotated supervised revision data, has inspired our investigation into the particular effects of combining error types in an artificial corpus.

One particularly interesting approach to generating artificial data is from Cahill et al. (2013), who, focusing on preposition errors, creates confusion sets for each preposition using supervised revision data, and selects replacements at random from these probability distributions. This approach was developed from Rozovskaya and Roth (2010), who first suggested the idea of probabilistically selecting likely error candidates. Interestingly, the artificial data proved to make manually annotated data more robust, meaning that it generalised better across different types of test sets, despite the fact that the overall quality of corrections was lowered. This was confirmed by Felice (2016), who also found that this kind of probabilistic error generation increases precision, and lowers recall.

One main focus of our research is the effect of the amount of errors in the training corpus on the amount of corrections made at test time. Rozovskaya et al. (2012) identify a useful technique known as error inflation, where more errors are introduced into the training data in order to improve recall. This is further explored in our work.

3 Experimental Setup

3.1 Data

In our research, errors are systemically introduced into "correct" English data. The correct data comes from the NewsCrawl corpus in WMT-2016.[1] It is open domain, featuring a wide variety of topics and writing styles, taken from recent articles. We used 21,789,157 sentences for training, and 5,447,288 held-out sentences from the same source for a development set.

We follow the same methodology of Cahill et al. (2013) to generate noise. Specifically, supervised revision data is used to see how often particular words are corrected into specific prepositions or determiners. The revision data which is used for our research is the Lang-8 corpus, which is available for academic purposes upon request.[2] The corpus is scraped from the Lang-8 website, where crowd-sourced grammar corrections are posted for non-native speakers of English. It is arguably more reliable than Wikipedia, which contains vandalism, however, it is noticeably smaller than Wikipedia.

The process of introducing errors into the WMT data using the Lang-8 corpus is as follows:

1. Extract plain text versions of the Lang-8 corpus, consisting solely of sentences with corrections

2. Compare source sentence with corrections using an efficient *diff* algorithm.[3] Note that this often included several steps of revisions.

3. Prepare a list of all prepositions/determiners. This is taken from the tags of the WMT data retrieved from the Stanford tagger.[4]

4. Remove all sentences that do not contain a single revision involving a preposition or a determiner. Using a hand-crafted set of possible prepositions/determiners, it is determined for each sentence whether it involves a deletion (eg. "for" → "NULL"), an addition (eg. "NULL" → "the"), or a replacement (eg. "on" → "in").

5. Generate confusion sets for each preposition/determiner by listing all the deletions which are replaced by that word, and counting the frequency of each specific revision.

[1] http://www.statmt.org/wmt16/translation-task.html

[2] http://cl.naist.jp/nldata/lang-8

[3] http://code.google.com/p/google-diff-match-patch

[4] Using word lists has the advantage of not relying on unsupervised POS-tagging methods. However, there are certain ambiguities which are not addressed. In this research, the preposition "to" is not included, due to confusion with the infinitive particle. There are however other less frequent ambiguous cases which are included, such as "that" and "before", which can both appear as conjunctions. Future experiments would benefit from a comparison of the performance of POS tags against word lists.

From there, generate a probability distribution for each preposition/determiner.

6. Insert the target word itself into the distribution with a frequency relative to the error rate. An 80% error rate for example means that 20% of the time, the same word is selected, effectively leaving it in its "correct" form.

7. Prepositions/determiners in the WMT corpus are systemically replaced by one of the options in their respective probability distributions, selected at random by a sampler.

3.2 Experiments

Cahill et al. (2013) have made their revision data extracted from Wikipedia available for download, which is why it is appropriate to compare it to the revision data which is extracted from Lang-8. Both sets of revision data are used to create two separate confusion sets for prepositions. They are then used to create two sets of error corpora in which 20%, 40%, 60% and 80% of prepositions are altered according to the error introduction procedure detailed above.

To compare, revision data extracted from Lang-8 is also used to create error corpora containing the same amounts of prepositional error. It is worth noting that Cahill et al.'s research does not include the empty "NULL" preposition, meaning that errors in which a preposition is missing are not accounted for. By contrast, in our work we include every case in which a preposition is inserted, as well as replaced, although we do not deal with deletions. Deleting prepositions which were inserted in the revision data simply follows the same procedure as replacements, where a preposition is replaced with the null preposition. Inserting prepositions which were deleted in the revision data is much more difficult, as it is not clear where in a sentence each preposition should be. The use of context words before and after a deletion is being explored in more current research, but does not feature in these experiments. This is nevertheless a major contribution, because insertions and deletions make up a significant part of the errors. In Lang-8, for example, there were 10054 corrections of prepositions, of which 4274 were insertions, and 2657 were deletions. This means that replacements only consist of 31% percent of the errors.

We also use determiner revision data extracted from Lang-8 to create determiner errors in a similar fashion, with 20%, 40%, 60% and 80% of errors.

A final set of synthetic error data is then generated where both prepositions and determiners are introduced into the same corpus, containing 20%, 40%, 60% and 80% of both kinds of error. This is to investigate whether the GEC system is capable of dealing with two types of error at once.

3.3 Evaluation

In order to test the effects of mismatching error density and type between training and test data, each model is tested on specially created test sets with varying amounts of error in them. Cahill et al. (2013) found that the highest scores came from models both trained and tested on similar error rates. Our research aims to build on this finding.

The first test set is made from Lang-8, which is also used to create the confusion sets for the training data. Specifically, only the sentences with prepositions, determiners, and a mix of both in the revisions are used. No other types of error are included. These sentences are mixed with corrected sentences (where the revised sentence is used as both source and target) to varying degrees. In each case, 1000 sentences of erroneous data are mixed with either 4000, 1500, 666, or 250 sentences of "correct" English, also taken from Lang-8. This is in order to create test sets in which 20%, 40%, 60%, and 80% of sentences are erroneous, similar to the training data. Table 1 shows the test sets created out of the Lang-8 corpus.

The NUCLE corpus (Ng et al., 2014) was used as training and test sets for the CoNLL-2014 Shared Task (Ng et al., 2014) on GEC, and since then has been commonly used in the field for comparison with previous work. The NUCLE corpus is used in our research in order to generate test sets from a different domain, despite those test sets being smaller. Again, prepositions, determiners and a combination of both are extracted and mixed with corrected sentences from the same corpus. Due to the smaller amount of relevant errors, as many sentences containing each error as possible are taken. For prepositions, this amounts 332 sentences, for determiners, 595 sentences, and for both, 169 sentences. Table 2 shows the test sets created out of the NUCLE corpus.

For our experiments we use OpenNMT, an open-source implementation of a bidirectional

Test sets	Error type	Error Rate	Size
test-l8p20	Preposition	20%	5000
test-l8p40	Preposition	40%	2500
test-l8p60	Preposition	60%	1666
test-l8p80	Preposition	80%	1250
test-l8d20	Determiner	20%	5000
test-l8d40	Determiner	40%	2500
test-l8d60	Determiner	60%	1666
test-l8d80	Determiner	80%	1250
test-l8b20	Both	20%	5000
test-l8b40	Both	40%	2500
test-l8b60	Both	60%	1666
test-l8b80	Both	80%	1250

Table 1: Lang-8 Corpus test sets

Test sets	Error type	Error Rate	Size
test-np20	Preposition	20%	1660
test-np40	Preposition	40%	830
test-np60	Preposition	60%	553
test-np80	Preposition	80%	415
test-nd20	Determiner	20%	2975
test-nd40	Determiner	40%	1487
test-nd60	Determiner	60%	992
test-nd80	Determiner	80%	744
test-nb20	Both	20%	845
test-nb40	Both	40%	423
test-nb60	Both	60%	282
test-nb80	Both	80%	211

Table 2: NUCLE Corpus test sets

RNN encoder–decoder network with attention[5]. OpenNMT was chosen because of its ease of use, and similarity to the architecture used by the current state of the art results reported by Yuan and Briscoe (2016). The selected evaluation metric is the GLEU score, which has been shown to be the most appropriate metric for GEC (Napoles et al., 2015).

4 Results and Discussion

The first objective of our research is to see the difference between testing on Lang-8 and NUCLE test sets when trained on data containing varying error densities created using data from Lang-8. For prepositional errors, the GLEU scores of the four different models are in Table 3, and the results are plotted in Figure 4. When tested on corpora with only 20% error, the GLEU score remains the same on both test sets. However, the higher the error rate in the test set, the better the models perform on the NUCLE set in comparison with the Lang-8 set. This is surprising, seeing as the Lang-8 corpus was used to inform the process of error generation in the training set.

In the tables cited in this paper, it is expected that the highest scores will occur along the diagonal. A test set containing 20% error would be best handled by training data which also contains 20% error. Likewise with 40%, 60% and 80 %. Conversely, training data containing 80% error would not perform as well on test data containing 40% as the training data which also has 40% error. This data shows, however that this is not always the case. When testing on 80% error, the models trained on 80% error density themselves obtain – as expected – the highest score, although only slightly. Interestingly, however, the 80% models also perform better on the 40% and 60% test sets, which seems to confirm Rozovskaya et al. (2012)'s "Error Inflation" idea. This is the idea that putting more errors than needed into the training data helps the model generalise more.

One interesting observation from the data is the fact that all the models perform better on the 20% test sets. This is likely because the models are capable of recognising that a sentence need not be corrected, and doing so is simpler than finding a correction of incorrect sentences.

Testing on determiner errors revealed similar results. The results are provided in Table 4, and plotted in Figure 4. In this case, error inflation does not seem to work, as the highest scoring results for each test set is more or less the training set with the matching error density. This indicates that systems that correct determiners have different properties to those which correct prepositions.

[5]http://opennmt.net/

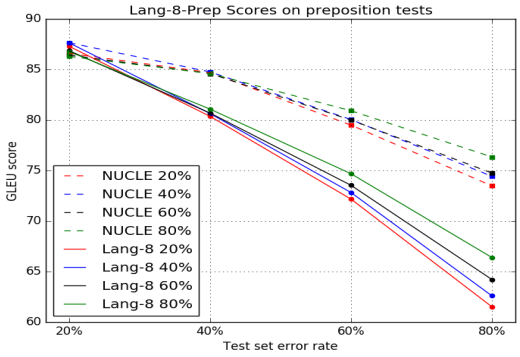

Figure 1: Plot of the data in Table 3

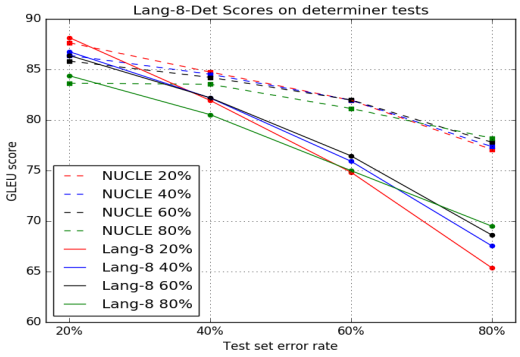

Figure 2: Plot of the data in Table 4

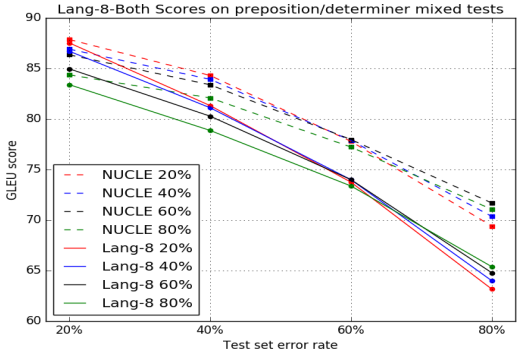

Figure 3: Plot of the data in Table 5

The results of training models on data containing a combination of both kinds of error on combined preposition and determiner test data is shown in Table 5 and Figure 4. The data is consists of slightly lower scores in general, suggesting that mixing error types does not have as high a quality of correction as single errors. Also, the NUCLE test scores in particular suffer in comparison with the singular error models, showing a failure to generalise across domains. Finally, "Error Inflation" also does not appear to work here.

These results shed doubt on the "Error Inflation" present in the preposition experiment. If it were dependent on the type of error, and prepositions were the kind which encouraged the use of "Error Inflation", then it follows that it should at least be present in the combined models. Instead of different error types subtly influencing the behaviour of the combined model in a cumulative way, the behaviour seems more random. In one case, the 20% combined model performs better on the 40% NUCLE test set than the 40% one, which suggests that reducing the amount of introduced error would make an improvement.

Table 6 and Figure 4 show how well the combined model performs on test sets with individual error types only. First of all, the scores are lower than the respective values attained by models trained on individual errors on the same test sets, but only slightly. Also, as seen in Tables 3, 4 and 5, the combined model testing on the combined test set returns lower scores than the individual models testing on their respective test sets with just one of the error types. However, the combined models' scores are better than those achieved by the individual models on the combined test sets, as shown in Table 7 and Figure 4. This indicates that the combined model is better suited for tackling both errors at once, and only a little worse at tackling individual errors than the individual error models. This is a predictable outcome, but the reduction in GLEU score suggests that combining errors in an attempt to correct all errors will generate noise, and the more error types that are covered, the less likely that they will be correctly revised at test time, which makes the idea of making an generalised corrector for all errors less feasible.

It is also worth mentioning that correcting determiners seems to result in higher scores than correcting prepositions. This could be due to the amount of possible prepositions that need to

be considered compared to the determiners. Although many determiners are considered, the vast majority of the cases involve the three articles "a", "an" and "the", as well as the null determiner. This is evidence for the need to consider the variation between different errors types when generating errors.

The final research question is whether the confusion set generated from Wikipedia revisions by Cahill et al. (2013) is much different from the one generated from Lang-8. Table 8 and Figure 4 show the results of preposition models informed by Wikipedia and Lang-8 tested on Lang-8 test sets. Table 9 and Figure 4 show the results of the same models on the NUCLE test sets. As expected, the errors generated from the confusion set informed by Lang-8 performs better on the Lang-8 test sets than on the NUCLE test sets. What is interesting, however, is that the Wikipedia revisions performed significantly better not only on the NUCLE test sets, but also on the Lang-8 test sets. This is surprising, because the Wikipedia revisions are not necessarily in the same domain, whereas the Lang-8 revisions are from the same dataset. Furthermore, the Wikipedia revisions do not take insertions or deletions into account. It is clear that the amount of revisions considered makes a difference: there were 10054 Lang-8 revisions, and 303847 Wikipedia revisions, 30 times more. The small amount of Lang-8 revisions could also account for the noise identified in the Lang-8 models, but this noise is also present in the Wikipedia revisions, where "error inflation" appears to only appear sometimes and not always.

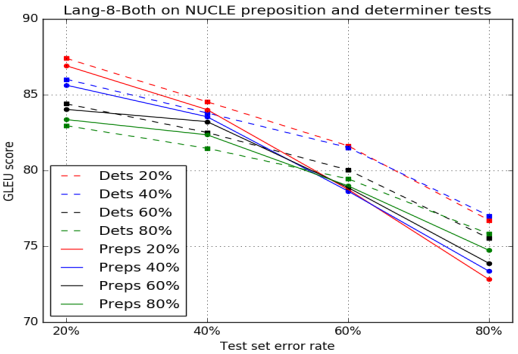

Figure 4: Plot of the data in Table 6

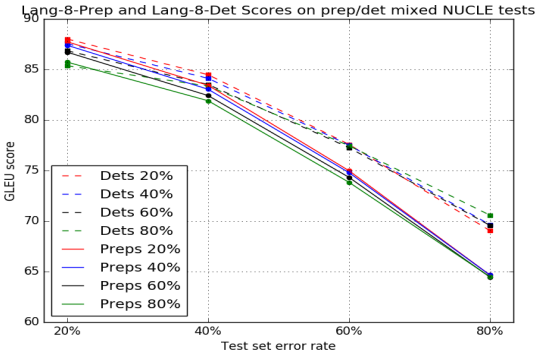

Figure 5: Plot of the data in Table 7

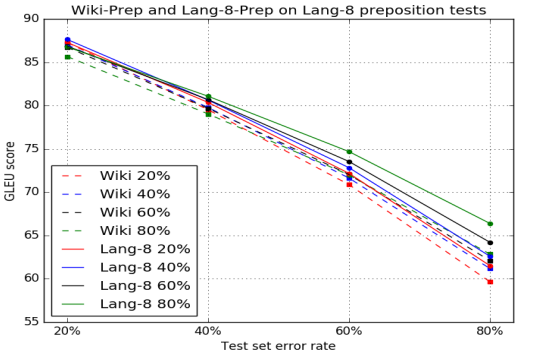

Figure 6: Plot of the data in Table 8

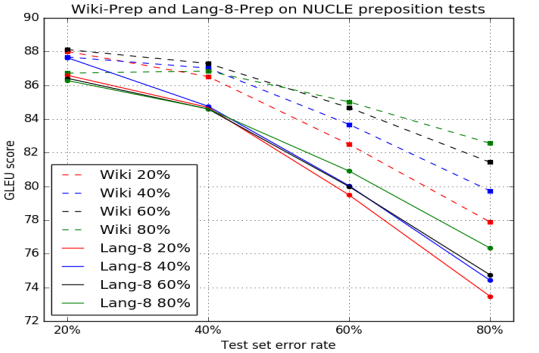

Figure 7: Plot of the data in Table 9

5 Conclusion

Our research aims to shed light on the issue of choosing how many errors to include in artificially generated erroneous data by tackling two specific error types. Results reveal some predictable outcomes, such as that it is easier to deal with test corpora which have smaller error rates, because leaving correct sentences alone is easier for the model to learn than making a good correction. Also, in most cases, there is a correlation between the error rate of the training data and the test data. However, some of the results revealed unexpected outcomes. Although it is possible that the data is noisy, the results, particularly for the prepositions, support a concept called "Error Inflation", which suggests that including more errors into the training data will lead to a higher GLEU score. This effect was not observed in the determiner and combined models, suggesting that there might be variation between different error types depending on the distribution of revisions made for that error type. It is possible to combine two error types together into one training set, and tackle two error types at once at test time, although the scores are not as high as when solving only individual errors. Also, the confusion set generated from Wikipedia revisions proved to yield better results than that generated from Lang-8, due to the significantly larger number of revisions. Finally, this research supports generating erroneous data as a valid approach to improving neural models for GEC, and informs future researchers about the effects of error rate mismatches in training and test data.

Acknowledgments

We would like to thank Joel Tetreault for his advice and experience in Automatic Error Generation, as well as Mamoru Komachi of Lang-8 for access to the Lang-8 dataset.

References

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using Wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia, pages 507–517. http://www.aclweb.org/anthology/N13-1055.

Mariano Felice. 2016. Artificial error generation for translation-based grammatical error correction. Technical Report UCAM-CL-TR-895, University of Cambridge, Computer Laboratory. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-895.pdf.

Zhuoran Liu and Yang Liu. 2016. Exploiting unlabeled data for neural grammatical error detection. *ArXiv preprint* 1611.08987. http://arxiv.org/abs/1611.08987.

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)*. 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*. Baltimore, Maryland, pages 1–14. http://www.aclweb.org/anthology/W14-1701.

Alla Rozovskaya and Dan Roth. 2010. 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*. Los Angeles, California, pages 154–162. http://www.aclweb.org/anthology/N10-1018.

Alla Rozovskaya, Mark Sammons, and Dan Roth. 2012. The UI system in the HOO 2012 shared task on error correction. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Montréal, Canada, pages 272–280. http://www.aclweb.org/anthology/W12-2032.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California, pages 380–386. http://www.aclweb.org/anthology/N16-1042.

A Appendix - Tables of results

Test Sets	20%	40%	60%	80%
test-l8p20	87.29	87.63	86.85	86.77
test-l8p40	80.35	80.63	80.68	81.07
test-l8p60	72.17	72.82	73.54	74.68
test-l8p80	61.51	62.60	64.21	66.39
test-np20	86.60	87.64	86.42	86.28
test-np40	84.69	84.76	84.59	84.60
test-np60	79.49	80.04	79.99	80.91
test-np80	73.48	74.43	74.75	76.34

Table 3: GLEU score according to how much **preposition** error in training data informed by Lang-8, tested on test sets with varying amounts of error from **Lang-8 and NUCLE**.

Test Sets	20%	40%	60%	80%
test-l8d20	88.16	86.76	86.39	84.38
test-l8d40	81.94	82.18	82.20	80.53
test-l8d60	74.83	75.92	76.45	74.99
test-l8d80	65.36	67.54	68.61	69.49
test-nd20	87.64	86.40	85.82	83.63
test-nd40	84.74	84.53	84.22	83.53
test-nd60	81.95	81.96	81.98	81.13
test-nd80	77.08	77.36	77.79	78.20

Table 4: GLEU score according to how much **determiner** error in training data informed by Lang-8, tested on test sets with varying amounts of error from **Lang-8 and NUCLE**.

Test Sets	20%	40%	60%	80%
test-l8b20	87.53	86.70	84.96	83.40
test-l8b40	81.32	81.11	80.28	78.86
test-l8b60	73.74	73.99	73.99	73.37
test-l8b80	63.16	63.98	64.77	65.38
test-nb20	87.86	86.91	86.35	84.36
test-nb40	84.30	83.92	83.38	82.09
test-nb60	77.78	77.87	77.96	77.22
test-nb80	69.39	70.35	71.68	71.05

Table 5: GLEU score according to how much **combined** preposition and determiner error in training data informed by Lang-8, tested on test sets with varying amounts of error from **Lang-8 and NUCLE**.

Test Sets	20%	40%	60%	80%
test-nd20	87.40	86.02	84.40	82.95
test-nd40	84.53	83.80	82.51	81.47
test-nd60	81.63	81.52	80.04	79.45
test-nd80	76.73	77.01	75.53	75.82
test-np20	86.91	85.63	84.03	83.36
test-np40	84.02	83.55	83.22	82.35
test-np60	78.75	78.64	78.88	78.99
test-np80	72.82	73.36	73.88	74.74

Table 6: GLEU score according to how much **combined** preposition and determiner error in training data informed by Lang-8, tested separately on **NUCLE** test sets with varying amounts of **determiner** error, and then **preposition** error.

Test Sets	20%	40%	60%	80%
test-nb20	88.01	87.55	86.85	85.36
test-nb40	84.48	84.12	83.51	83.36
test-nb60	77.58	77.51	77.26	77.45
test-nb80	69.05	69.64	69.57	70.58
test-nb20	87.74	87.39	86.71	85.72
test-nb40	83.43	83.04	82.41	81.90
test-nb60	74.98	74.79	74.32	73.83
test-nb80	64.66	64.66	64.45	64.49

Table 7: GLEU score according to how much **preposition** error (first 4 rows) or **determiner** error (last 4 rows) in training data informed by Lang-8, tested on test sets with varying amounts of **combined** determiner/preposition error from NUCLE.

Test Sets	20%	40%	60%	80%
test-l8p20	87.27	87.00	86.72	85.66
test-l8p40	79.52	79.78	79.67	79.03
test-l8p60	70.87	71.63	72.03	71.99
test-l8p80	59.66	61.18	62.09	62.85
test-l8p20	87.29	87.63	86.85	86.77
test-l8p40	80.35	80.63	80.68	81.07
test-l8p60	72.17	72.82	73.54	74.68
test-l8p80	61.51	62.60	64.21	66.39

Table 8: GLEU score according to how much preposition error in training data informed by **Wikipedia** (first 4 rows) and **Lang-8** (last 4 rows), tested on test sets with varying amounts of preposition error from **Lang-8**.

Test Sets	20%	40%	60%	80%
test-np20	88.01	87.68	88.12	86.72
test-np40	86.52	87.02	87.29	86.84
test-np60	82.49	83.68	84.67	85.02
test-np80	77.89	79.74	81.43	82.56
test-np20	86.60	87.64	86.42	86.28
test-np40	84.69	84.76	84.59	84.60
test-np60	79.49	80.04	79.99	80.91
test-np80	73.48	74.43	74.75	76.34

Table 9: GLEU score according to how much preposition error in training data informed by **Wikipedia** (first 4 rows) and **Lang-8** (last 4 rows), tested on test sets with varying amounts of preposition error from **NUCLE**.