Artificial Error Generation with Machine Translation and Syntactic Patterns

Marek Rei **Mariano Felice** **Zheng Yuan** **Ted Briscoe**
The ALTA Institute
Computer Laboratory
University of Cambridge
United Kingdom
firstname.lastname@cl.cam.ac.uk

Abstract

Shortage of available training data is holding back progress in the area of automated error detection. This paper investigates two alternative methods for artificially generating writing errors, in order to create additional resources. We propose treating error generation as a machine translation task, where grammatically correct text is translated to contain errors. In addition, we explore a system for extracting textual patterns from an annotated corpus, which can then be used to insert errors into grammatically correct sentences. Our experiments show that the inclusion of artificially generated errors significantly improves error detection accuracy on both FCE and CoNLL 2014 datasets.

1 Introduction

Writing errors can occur in many different forms – from relatively simple punctuation and determiner errors, to mistakes including word tense and form, incorrect collocations and erroneous idioms. Automatically identifying all of these errors is a challenging task, especially as the amount of available annotated data is very limited. Rei and Yannakoudakis (2016) showed that while some error detection algorithms perform better than others, it is additional training data that has the biggest impact on improving performance.

Being able to generate realistic artificial data would allow for any grammatically correct text to be transformed into annotated examples containing writing errors, producing large amounts of additional training examples. Supervised error generation systems would also provide an efficient method for anonymising the source corpus – error statistics from a private corpus can be aggregated and applied to a different target text, obscuring sensitive information in the original examination scripts. However, the task of creating incorrect data is somewhat more difficult than might initially appear – naive methods for error generation can create data that does not resemble natural errors, thereby making downstream systems learn misleading or uninformative patterns.

Previous work on artificial error generation (AEG) has focused on specific error types, such as prepositions and determiners (Rozovskaya and Roth, 2010, 2011), or noun number errors (Brockett et al., 2006). Felice and Yuan (2014) investigated the use of linguistic information when generating artificial data for error correction, but also restricting the approach to only five error types. There has been very limited research on generating artificial data for all types, which is important for general-purpose error detection systems. For example, the error types investigated by Felice and Yuan (2014) cover only 35.74% of all errors present in the CoNLL 2014 training dataset, providing no additional information for the majority of errors.

In this paper, we investigate two supervised approaches for generating all types of artificial errors. We propose a framework for generating errors based on statistical machine translation (SMT), training a model to translate from correct into incorrect sentences. In addition, we describe a method for learning error patterns from an annotated corpus and transplanting them into error-free text. We evaluate the effect of introducing artificial data on two error detection benchmarks. Our results show that each method provides significant improvements over using only the available training set, and a combination of both gives an absolute improvement of 4.3% in $F_{0.5}$, without requiring any additional annotated data.

Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 287–292
Copenhagen, Denmark, September 8, 2017.

Original	We are a well-mixed class with equal numbers of boys and girls, all about 20 years old.
FY14	We **am** a well-mixed class with equal numbers of boys and girls, all about 20 years old.
PAT	We are a well-mixed class with equal numbers of boys **an** girls, all about 20 **year** old.
MT	We are a well-mixed class with **equals** numbers of boys and girls, all about 20 years old.

Table 1: Example artificial errors generated by three systems: the error generation method by Felice and Yuan (2014) (FY14), our pattern-based method covering all error types (PAT), and the machine translation approach to artificial error generation (MT).

2 Error Generation Methods

We investigate two alternative methods for AEG. The models receive grammatically correct text as input and modify certain tokens to produce incorrect sequences. The alternative versions of each sentence are aligned using Levenshtein distance, allowing us to identify specific words that need to be marked as errors. While these alignments are not always perfect, we found them to be sufficient for practical purposes, since alternative alignments of similar sentences often result in the same binary labeling. Future work could explore more advanced alignment methods, such as proposed by Felice et al. (2016).

In Section 4, this automatically labeled data is then used for training error detection models.

2.1 Machine Translation

We treat AEG as a translation task – given a correct sentence as input, the system would learn to translate it to contain likely errors, based on a training corpus of parallel data. Existing SMT approaches are already optimised for identifying context patterns that correspond to specific output sequences, which is also required for generating human-like errors. The reverse of this idea, translating from incorrect to correct sentences, has been shown to work well for error correction tasks (Brockett et al., 2006; Ng et al., 2014), and round-trip translation has also been shown to be promising for correcting grammatical errors (Madnani et al., 2012).

Following previous work (Brockett et al., 2006; Yuan and Felice, 2013), we build a phrase-based SMT error generation system. During training, error-corrected sentences in the training data are treated as the source, and the original sentences written by language learners as the target. Pialign (Neubig et al., 2011) is used to create a phrase translation table directly from model probabilities. In addition to default features, we add character-level Levenshtein distance to each mapping in the phrase table, as proposed by Felice et al. (2014). Decoding is performed using Moses (Koehn et al., 2007) and the language model used during decoding is built from the original erroneous sentences in the learner corpus. The IRSTLM Toolkit (Federico et al., 2008) is used for building a 5-gram language model with modified Kneser-Ney smoothing (Kneser and Ney, 1995).

2.2 Pattern Extraction

We also describe a method for AEG using patterns over words and part-of-speech (POS) tags, extracting known incorrect sequences from a corpus of annotated corrections. This approach is based on the best method identified by Felice and Yuan (2014), using error type distributions; while they covered only 5 error types, we relax this restriction and learn patterns for generating all types of errors.

The original and corrected sentences in the corpus are aligned and used to identify short transformation patterns in the form of *(incorrect phrase, correct phrase)*. The length of each pattern is the affected phrase, plus up to one token of context on both sides. If a word form changes between the incorrect and correct text, it is fully saved in the pattern, otherwise the POS tags are used for matching.

For example, the original sentence '*We went shop on Saturday*' and the corrected version '*We went shopping on Saturday*' would produce the following pattern:

(VVD shop_VV0 II, VVD shopping_VVG II)

After collecting statistics from the background corpus, errors can be inserted into error-free text. The learned patterns are now reversed, looking for the correct side of the tuple in the input sentence. We only use patterns with frequency $>= 5$, which yields a total of 35,625 patterns from our training data. For each input sentence, we first decide how many errors will be generated (using probabilities from the background corpus) and attempt to cre-

ate them by sampling from the collection of applicable patterns. This process is repeated until all the required errors have been generated or the sentence is exhausted. During generation, we try to balance the distribution of error types as well as keeping the same proportion of incorrect and correct sentences as in the background corpus (Felice, 2016). The required POS tags were generated with RASP (Briscoe et al., 2006), using the CLAWS2 tagset.

3 Error Detection Model

We construct a neural sequence labeling model for error detection, following the previous work (Rei and Yannakoudakis, 2016; Rei, 2017). The model receives a sequence of tokens as input and outputs a prediction for each position, indicating whether the token is correct or incorrect in the current context. The tokens are first mapped to a distributed vector space, resulting in a sequence of word embeddings. Next, the embeddings are given as input to a bidirectional LSTM (Hochreiter and Schmidhuber, 1997), in order to create context-dependent representations for every token. The hidden states from forward- and backward-LSTMs are concatenated for each word position, resulting in representations that are conditioned on the whole sequence. This concatenated vector is then passed through an additional feedforward layer, and a softmax over the two possible labels (*correct* and *incorrect*) is used to output a probability distribution for each token. The model is optimised by minimising categorical cross-entropy with respect to the correct labels. We use AdaDelta (Zeiler, 2012) for calculating an adaptive learning rate during training, which accounts for a higher baseline performance compared to previous results.

4 Evaluation

We trained our error generation models on the public FCE training set (Yannakoudakis et al., 2011) and used them to generate additional artificial training data. Grammatically correct text is needed as the starting point for inserting artificial errors, and we used two different sources: 1) the corrected version of the same FCE training set on which the system is trained (450K tokens), and 2) example sentences extracted from the English Vocabulary Profile (270K tokens).[1]. While there are other text corpora that could be used (e.g., Wikipedia and news articles), our development experiments showed that keeping the writing style and vocabulary close to the target domain gives better results compared to simply including more data.

We evaluated our detection models on three benchmarks: the FCE test data (41K tokens) and the two alternative annotations of the CoNLL 2014 Shared Task dataset (30K tokens) (Ng et al., 2014). Each artificial error generation system was used to generate 3 different versions of the artificial data, which were then combined with the original annotated dataset and used for training an error detection system. Table 1 contains example sentences from the error generation systems, highlighting each of the edits that are marked as errors.

The error detection results can be seen in Table 2. We use $F_{0.5}$ as the main evaluation measure, which was established as the preferred measure for error correction and detection by the CoNLL-14 shared task (Ng et al., 2014). $F_{0.5}$ calculates a weighted harmonic mean of precision and recall, which assigns twice as much importance to precision – this is motivated by practical applications, where accurate predictions from an error detection system are more important compared to coverage. For comparison, we also report the performance of the error detection system by Rei and Yannakoudakis (2016), trained using the same FCE dataset.

The results show that error detection performance is substantially improved by making use of artificially generated data, created by any of the described methods. When comparing the error generation system by Felice and Yuan (2014) (FY14) with our pattern-based (PAT) and machine translation (MT) approaches, we see that the latter methods covering all error types consistently improve performance. While the added error types tend to be less frequent and more complicated to capture, the added coverage is indeed beneficial for error detection. Combining the pattern-based approach with the machine translation system (Ann+PAT+MT) gave the best overall performance on all datasets. The two frameworks learn to generate different types of errors, and taking advantage of both leads to substantial improvements in error detection.

We used the Approximate Randomisation Test (Noreen, 1989; Cohen, 1995) to calculate statistical significance and found that the improvement

[1] http://www.englishprofile.org/wordlists

	FCE			CoNLL-14 TEST1			CoNLL-14 TEST2		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
R&Y (2016)	46.10	28.50	41.10	15.40	**22.80**	16.40	23.60	**25.10**	23.90
Annotation	53.91	26.88	44.84	16.12	18.42	16.52	25.72	20.92	24.57
Ann+FY14	58.77	25.55	46.54	20.48	14.41	18.88	33.25	16.67	27.72
Ann+PAT	**62.47**	24.70	47.81	21.07	15.02	19.47	34.04	17.32	28.49
Ann+MT	58.38	**28.84**	48.37	19.52	20.79	19.73	30.24	22.96	28.39
Ann+PAT+MT	60.67	28.08	**49.11**	**23.28**	18.01	**21.87**	**35.28**	19.42	**30.13**

Table 2: Error detection performance when combining manually annotated and artificial training data.

for each of the systems using artificial data was significant over using only manual annotation. In addition, the final combination system is also significantly better compared to the Felice and Yuan (2014) system, on all three datasets. While Rei and Yannakoudakis (2016) also report separate experiments that achieve even higher performance, these models were trained on a considerably larger proprietary corpus. In this paper we compare error detection frameworks trained on the same publicly available FCE dataset, thereby removing the confounding factor of dataset size and only focusing on the model architectures.

The error generation methods can generate alternative versions of the same input text – the pattern-based method randomly samples the error locations, and the SMT system can provide an n-best list of alternative translations. Therefore, we also investigated the combination of multiple error-generated versions of the input files when training error detection models. Figure 1 shows the $F_{0.5}$ score on the development set, as the training data is increased by using more translations from the n-best list of the SMT system. These results reveal that allowing the model to see multiple alternative versions of the same file gives a distinct improvement – showing the model both correct and incorrect variations of the same sentences likely assists in learning a discriminative model.

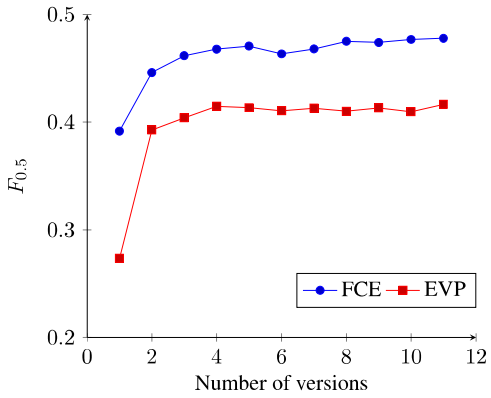

Figure 1: $F_{0.5}$ on FCE development set with increasing amounts of artificial data from SMT.

5 Related Work

Our work builds on prior research into AEG. Brockett et al. (2006) constructed regular expressions for transforming correct sentences to contain noun number errors. Rozovskaya and Roth (2010) learned confusion sets from an annotated corpus in order to generate preposition errors. Foster and Andersen (2009) devised a tool for generating errors for different types using patterns provided by the user or collected automatically from an annotated corpus. However, their method uses a limited number of edit operations and is thus unable to generate complex errors. Cahill et al. (2013) compared different training methodologies and showed that artificial errors helped correct prepositions. Felice and Yuan (2014) learned error type distributions for generating five types of errors, and the system in Section 2.2 is an extension of this model. While previous work focused on generating a specific subset of error types, we explored two holistic approaches to AEG and showed that they are able to significantly improve error detection performance.

6 Conclusion

This paper investigated two AEG methods, in order to create additional training data for error detection. First, we explored a method using textual patterns learned from an annotated corpus, which are used for inserting errors into correct input text. In addition, we proposed formulating error generation as an MT framework, learning to translate from grammatically correct to incorrect sentences.

The addition of artificial data to the training process was evaluated on three error detection anno-

tations, using the FCE and CoNLL 2014 datasets. Making use of artificial data provided improvements for all data generation methods. By relaxing the type restrictions and generating all types of errors, our pattern-based method consistently outperformed the system by Felice and Yuan (2014). The combination of the pattern-based method with the machine translation approach gave further substantial improvements and the best performance on all datasets.

References

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The Second Release of the RASP System. In *Proceedings of the COLING/ACL on Interactive presentation sessions.*. Association for Computational Linguistics, Sydney, Australia, July, pages 77–80. https://doi.org/10.3115/1225403.1225423.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* https://doi.org/10.3115/1220175.1220207.

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*. http://www.aclweb.org/anthology/N13-1055.

Paul Cohen. 1995. *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, MA.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association*.

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.

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 and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)* .

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 18th Conference on Computational Natural Language Learning: Shared Task*.

Jennifer Foster and Øistein E. Andersen. 2009. GenERRate: generating errors for use in grammatical error detection. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*. http://dl.acm.org/citation.cfm?id=1609855.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation* 9. https://doi.org/10.1.1.56.7752.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*.

Nitin Madnani, 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*. Association for Computational Linguistics, Montréal, Canada, pages 44–53. http://www.aclweb.org/anthology/W12-2005.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

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*. http://www.aclweb.org/anthology/W/W14/W14-1701.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses*. Wiley, New York.

Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL-2017)*.

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*. https://aclweb.org/anthology/P/P16/P16-1112.pdf.

Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* http://dl.acm.org/citation.cfm?id=1870752.

Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)* http://www.aclweb.org/anthology-new/P/P11/P11-1093.pdf.

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*. http://www.aclweb.org/anthology/P11-1019.

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the 17th Conference on Computational Natural Language Learning: Shared Task*.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701* http://arxiv.org/abs/1212.5701.